

A Uniform Substitution Calculus for Differential Dynamic Logic

André Platzer*

July 31, 2015

Abstract

This paper introduces a new proof calculus for *differential dynamic logic* ($\text{d}\mathcal{L}$) that is entirely based on *uniform substitution*, a proof rule that substitutes a formula for a predicate symbol everywhere. Uniform substitutions make it possible to rely on *axioms* rather than axiom schemata, substantially simplifying implementations. Instead of subtle schema variables and soundness-critical side conditions on the occurrence patterns of variables, the resulting calculus adopts only a finite number of ordinary $\text{d}\mathcal{L}$ formulas as axioms. The static semantics of differential dynamic logic is captured exclusively in uniform substitutions and bound variable renamings as opposed to being spread in delicate ways across the prover implementation. In addition to sound uniform substitutions, this paper introduces *differential forms* for differential dynamic logic that make it possible to internalize differential invariants, differential substitutions, and derivations as first-class axioms in $\text{d}\mathcal{L}$.

Keywords: differential dynamic logic, uniform substitution, axioms, differentials, static semantics

1 Introduction

Differential dynamic logic ($\text{d}\mathcal{L}$) [5, 7] is a logic for proving correctness properties of hybrid systems. It has a sound and complete proof calculus relative to differential equations [5, 7] and a sound and complete proof calculus relative to discrete systems [7]. Both sequent calculi [5] and Hilbert-type axiomatizations [7] have been presented for $\text{d}\mathcal{L}$ but only the former has been implemented. The implementation of $\text{d}\mathcal{L}$'s sequent calculus in KeYmaera [11] makes it straightforward for users to prove properties of hybrid systems, because it provides rules performing natural decompositions for each operator. The downside is that the implementation of the rule schemata and their side conditions on occurrence constraints and relations of reading and writing of variables as well as rule applications in context is nontrivial and inflexible in KeYmaera.

*Computer Science Department, Carnegie Mellon University, Pittsburgh, USA aplatzer@cs.cmu.edu

The goal of this paper is to identify how to make it straightforward to implement the axioms and proof rules of differential dynamic logic by writing down a finite list of *axioms* (concrete formulas, not axiom schemata that represent an infinite list of axioms subject to sophisticated soundness-critical schema variable matching implementations). They require multiple axioms to be combined with one another to obtain the effect that a user would want for proving a hybrid system conjecture. This paper argues that this is still a net win for hybrid systems, because a substantially simpler prover core is easier to implement correctly, and the need to combine multiple axioms to obtain user-level proof steps can be achieved equally well by appropriate tactics, which are not soundness-critical.

To achieve this goal, this paper follows observations for differential game logic [9] that highlight the significance and elegance of *uniform substitutions*, a classical proof rule for first-order logic [2, §35,40]. Uniform substitutions uniformly instantiate predicate and function symbols with formulas and terms, respectively, as functions of their arguments. In the presence of the nontrivial binding structure that nondeterminism and differential equations of hybrid programs induce for the dynamic modalities of differential dynamic logic, flexible but sound uniform substitutions become more complex for \mathbf{dL} , but can still be read off elegantly from its static semantics. In fact, \mathbf{dL} 's static semantics is solely captured¹ in the implementation of uniform substitution (and bound variable renaming), thereby leading to a completely modular proof calculus.

This paper introduces a static and dynamic semantics for *differential-form* \mathbf{dL} , proves coincidence lemmas and uniform substitution lemmas, culminating in a soundness proof for uniform substitutions (Section 3). It exploits the new *differential forms* that this paper adds to \mathbf{dL} for internalizing differential invariants [6], differential cuts [6, 8], differential ghosts [8], differential substitutions, total differentials and Lie-derivations [6, 8] as first-class citizens in \mathbf{dL} , culminating in entirely modular axioms for differential equations and a superbly modular soundness proof (Section 4). This approach is to be contrasted with earlier approaches for differential invariants that were based on complex built-in rules [6, 8]. The relationship to related work from previous presentations of differential dynamic logic [5, 7] continues to apply except that \mathbf{dL} now internalizes differential equation reasoning axiomatically via differential forms.

2 Differential-Form Differential Dynamic Logic

2.1 Syntax

Formulas and hybrid programs (HPs) of \mathbf{dL} are defined by simultaneous induction based on the following definition of terms. Similar simultaneous inductions are used throughout the proofs for \mathbf{dL} . The set of all *variables* is \mathcal{V} . For any $V \subseteq \mathcal{V}$ is $V' \stackrel{\text{def}}{=} \{x' : x \in V\}$ the set of *differential symbols* x' for the variables in V . Function symbols are written f, g, h , predicate symbols p, q, r , and variables $x, y, z \in \mathcal{V}$ with differential symbols $x', y', z' \in \mathcal{V}'$. Program constants are a, b, c .

¹ This approach is dual to other successful ways of solving the intricacies and subtleties of substitutions [1, 3] by imposing occurrence side conditions on axiom schemata and proof rules, which is what uniform substitutions can get rid of.

Definition 1 (Terms). *Terms* are defined by this grammar (with $\theta, \eta, \theta_1, \dots, \theta_k$ as terms, $x \in \mathcal{V}$ as variable, $x' \in \mathcal{V}'$ differential symbol, and f function symbol):

$$\theta, \eta ::= x \mid x' \mid f(\theta_1, \dots, \theta_k) \mid \theta + \eta \mid \theta \cdot \eta \mid (\theta)'$$

Number literals such as 0,1 are allowed as function symbols without arguments that are always interpreted as the numbers they denote. Beyond differential symbols x' , *differential-form dL* allows *differentials* $(\theta)'$ of terms θ as terms for the purpose of axiomatically internalizing reasoning about differential equations.

Definition 2 (Hybrid program). *Hybrid programs* (HPs) are defined by the following grammar (with α, β as HPs, program constant a , variable x , term θ possibly containing x , and formula ψ of first-order logic of real arithmetic):

$$\alpha, \beta ::= a \mid x := \theta \mid x' := \theta \mid ?\psi \mid x' = \theta \& \psi \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

Assignments $x := \theta$ of θ to variable x , *tests* $?\psi$ of the formula ψ in the current state, *differential equations* $x' = \theta \& \psi$ restricted to the evolution domain constraint ψ , *nondeterministic choices* $\alpha \cup \beta$, *sequential compositions* $\alpha; \beta$, and *nondeterministic repetition* α^* are as usual in dL [5, 7]. The effect of the *differential assignment* $x' := \theta$ to differential symbol x' is similar to the effect of the assignment $x := \theta$ to variable x , except that it changes the value of the differential symbol x' around instead of the value of x . It is not to be confused with the differential equation $x' = \theta$, which will follow said differential equation continuously for an arbitrary amount of time. The differential assignment $x' := \theta$, instead, only assigns the value of θ to the differential symbol x' discretely once at an instant of time. Program constants a are uninterpreted, i.e. their behavior depends on the interpretation in the same way that the values of function symbols f and predicate symbols p depends on their interpretation.

Definition 3 (dL formula). The *formulas of (differential-form) differential dynamic logic (dL)* are defined by the grammar (with dL formulas ϕ, ψ , terms $\theta, \eta, \theta_1, \dots, \theta_k$, predicate symbol p , quantifier symbol C , variable x , HP α):

$$\phi, \psi ::= \theta \geq \eta \mid p(\theta_1, \dots, \theta_k) \mid C(\phi) \mid \neg\phi \mid \phi \wedge \psi \mid \forall x \phi \mid \exists x \phi \mid [\alpha]\phi \mid \langle \alpha \rangle \phi$$

Operators $>, \leq, <, \vee, \rightarrow, \leftrightarrow$ are definable, e.g., $\phi \rightarrow \psi$ as $\neg(\phi \wedge \neg\psi)$. Likewise $[\alpha]\phi$ is equivalent to $\neg\langle \alpha \rangle \neg\phi$ and $\forall x \phi$ equivalent to $\neg\exists x \neg\phi$. The modal formula $[\alpha]\phi$ expresses that ϕ holds after all runs of α , while the dual $\langle \alpha \rangle \phi$ expresses that there is a run of α after which ϕ holds. *Quantifier symbols* C (with formula ϕ as argument), i.e. higher-order predicate symbols that bind all variables of ϕ , are unnecessary but internalize contextual congruence reasoning efficiently.

2.2 Dynamic Semantics

A state is a mapping from variables \mathcal{V} and differential symbols \mathcal{V}' to \mathbb{R} . The set of states is denoted \mathcal{S} . Let ν_x^r denote the state that agrees with state ν except for the value of variable x , which is changed to $r \in \mathbb{R}$, and accordingly for $\nu_{x'}^r$. The interpretation of a function symbol f with arity n (i.e. with n arguments) is a smooth function $I(f) : \mathbb{R}^n \rightarrow \mathbb{R}$ of n arguments.

Definition 4 (Semantics of terms). For each interpretation I , the *semantics of a term* θ in a state $\nu \in \mathcal{S}$ is its value in \mathbb{R} . It is defined inductively as follows

1. $\llbracket x \rrbracket^I \nu = \nu(x)$ for variable $x \in \mathcal{V}$
2. $\llbracket x' \rrbracket^I \nu = \nu(x')$ for differential symbol $x' \in \mathcal{V}'$
3. $\llbracket f(\theta_1, \dots, \theta_k) \rrbracket^I \nu = I(f)(\llbracket \theta_1 \rrbracket^I \nu, \dots, \llbracket \theta_k \rrbracket^I \nu)$ for function symbol f
4. $\llbracket \theta + \eta \rrbracket^I \nu = \llbracket \theta \rrbracket^I \nu + \llbracket \eta \rrbracket^I \nu$
5. $\llbracket \theta \cdot \eta \rrbracket^I \nu = \llbracket \theta \rrbracket^I \nu \cdot \llbracket \eta \rrbracket^I \nu$
6. $\llbracket (\theta)' \rrbracket^I \nu = \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I}{\partial x}(\nu) = \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X}$

Time-derivatives are undefined in an isolated state ν . The clou is that differentials can still be given a local semantics: $\llbracket (\theta)' \rrbracket^I \nu$ is the sum of all (analytic) spatial partial derivatives of the value of θ by all variables x (or rather their values X) multiplied by the corresponding tangent described by the value $\nu(x')$ of differential symbol x' . That sum over all variables $x \in \mathcal{V}$ has finite support, because θ only mentions finitely many variables x and the partial derivative by variables x that do not occur in θ is 0. The spatial derivatives exist since $\llbracket \theta \rrbracket^I \nu$ is a composition of smooth functions, so smooth. Thus, the semantics of $\llbracket (\theta)' \rrbracket^I \nu$ is the *differential*² of (the value of) θ , hence a differential one-form giving a real value for each tangent vector (i.e. vector field) described by the values $\nu(x')$. The values $\nu(x')$ of the differential symbols x' describe an arbitrary tangent vector or vector field. Along the flow of (the vector field of a) differential equation, though, the value of the differential $(\theta)'$ coincides with the analytic time-derivative of θ (Lemma 11). The interpretation of predicate symbol p with arity n is an n -ary relation $I(p) \subseteq \mathbb{R}^n$. The interpretation of quantifier symbol C is a functional $I(C)$ mapping subsets $M \subseteq \mathcal{S}$ to subsets $I(C)(M) \subseteq \mathcal{S}$.

Definition 5 (d \mathcal{L} semantics). The *semantics of a d \mathcal{L} formula* ϕ , for each interpretation I with a corresponding set of states \mathcal{S} , is the subset $\llbracket \phi \rrbracket^I \subseteq \mathcal{S}$ of states in which ϕ is true. It is defined inductively as follows

1. $\llbracket \theta \geq \eta \rrbracket^I = \{\nu \in \mathcal{S} : \llbracket \theta \rrbracket^I \nu \geq \llbracket \eta \rrbracket^I \nu\}$
2. $\llbracket p(\theta_1, \dots, \theta_k) \rrbracket^I = \{\nu \in \mathcal{S} : (\llbracket \theta_1 \rrbracket^I \nu, \dots, \llbracket \theta_k \rrbracket^I \nu) \in I(p)\}$
3. $\llbracket C(\phi) \rrbracket^I = I(C)(\llbracket \phi \rrbracket^I)$ for quantifier symbol C
4. $\llbracket \neg \phi \rrbracket^I = (\llbracket \phi \rrbracket^I)^c = \mathcal{S} \setminus \llbracket \phi \rrbracket^I$
5. $\llbracket \phi \wedge \psi \rrbracket^I = \llbracket \phi \rrbracket^I \cap \llbracket \psi \rrbracket^I$

²A slight abuse of notation rewrites the differential as $\llbracket (\theta)' \rrbracket^I = d\llbracket \theta \rrbracket^I = \sum_{i=1}^n \frac{\partial \llbracket \theta \rrbracket^I}{\partial x^i} dx^i$ when x^1, \dots, x^n are the variables in θ and their differentials dx^i form the basis of the cotangent space, which, when evaluated at a point ν whose values $\nu(x')$ determine the tangent vector alias vector field, coincides with Def. 4.

6. $\llbracket \exists x \phi \rrbracket^I = \{\nu \in \mathcal{S} : \nu_x^r \in \llbracket \phi \rrbracket^I \text{ for some } r \in \mathbb{R}\}$
7. $\llbracket \langle \alpha \rangle \phi \rrbracket^I = \llbracket \alpha \rrbracket^I \circ \llbracket \phi \rrbracket^I = \{\nu : \omega \in \llbracket \phi \rrbracket^I \text{ for some } \omega \text{ such that } (\nu, \omega) \in \llbracket \alpha \rrbracket^I\}$
8. $\llbracket [\alpha] \phi \rrbracket^I = \llbracket \neg \langle \alpha \rangle \neg \phi \rrbracket^I = \{\nu : \omega \in \llbracket \phi \rrbracket^I \text{ for all } \omega \text{ such that } (\nu, \omega) \in \llbracket \alpha \rrbracket^I\}$

A \mathbf{dL} formula ϕ is *valid in I* , written $I \models \phi$, iff $\llbracket \phi \rrbracket^I = \mathcal{S}$, i.e. $\nu \in \llbracket \phi \rrbracket^I$ for all ν . Formula ϕ is *valid*, written $\models \phi$, iff $I \models \phi$ for all interpretations I .

The interpretation of a program constant a is a state-transition relation $I(a) \subseteq \mathcal{S} \times \mathcal{S}$, where $(\nu, \omega) \in I(a)$ iff a can run from initial state ν to final state ω .

Definition 6 (Transition semantics of HPs). For each interpretation I , each HP α is interpreted semantically as a binary transition relation $\llbracket \alpha \rrbracket^I \subseteq \mathcal{S} \times \mathcal{S}$ on states, defined inductively by

1. $\llbracket a \rrbracket^I = I(a)$ for program constants a
2. $\llbracket x := \theta \rrbracket^I = \{(\nu, \nu_x^r) : r = \llbracket \theta \rrbracket^I \nu\} = \{(\nu, \omega) : \omega = \nu \text{ except } \llbracket x \rrbracket^I \omega = \llbracket \theta \rrbracket^I \nu\}$
3. $\llbracket x' := \theta \rrbracket^I = \{(\nu, \nu_{x'}^r) : r = \llbracket \theta \rrbracket^I \nu\} = \{(\nu, \omega) : \omega = \nu \text{ except } \llbracket x' \rrbracket^I \omega = \llbracket \theta \rrbracket^I \nu\}$
4. $\llbracket ?\psi \rrbracket^I = \{(\nu, \nu) : \nu \in \llbracket \psi \rrbracket^I\}$
5. $\llbracket x' = \theta \& \psi \rrbracket^I = \{(\nu, \omega) : I, \varphi \models x' = \theta \wedge \psi, \text{ i.e. } \varphi(\zeta) \in \llbracket x' = \theta \wedge \psi \rrbracket^I \text{ for all } 0 \leq \zeta \leq r, \text{ for some function } \varphi : [0, r] \rightarrow \mathcal{S} \text{ of some duration } r \text{ for which all } \varphi(\zeta)(x') = \frac{d\varphi(t)(x)}{dt}(\zeta) \text{ exist and } \nu = \varphi(0) \text{ on } \{x'\}^c \text{ and } \omega = \varphi(r)\}; \text{ i.e., } \varphi \text{ solves the differential equation and satisfies } \psi \text{ at all times. In case } r = 0, \text{ the only condition is that } \nu = \omega \text{ on } \{x'\}^c \text{ and } \omega(x') = \llbracket \theta \rrbracket^I \nu \text{ and } \omega \in \llbracket \psi \rrbracket^I.$
6. $\llbracket \alpha \cup \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \cup \llbracket \beta \rrbracket^I$
7. $\llbracket \alpha; \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \circ \llbracket \beta \rrbracket^I = \{(\nu, \omega) : (\nu, \mu) \in \llbracket \alpha \rrbracket^I, (\mu, \omega) \in \llbracket \beta \rrbracket^I\}$
8. $\llbracket \alpha^* \rrbracket^I = (\llbracket \alpha \rrbracket^I)^* = \bigcup_{n \in \mathbb{N}} \llbracket \alpha^n \rrbracket^I$ with $\alpha^{n+1} \equiv \alpha^n; \alpha$ and $\alpha^0 \equiv ?\text{true}$

where ρ^* denotes the reflexive transitive closure of relation ρ .

The initial values $\nu(x')$ of differential symbols x' do *not* influence the behavior of $(\nu, \omega) \in \llbracket x' = \theta \& \psi \rrbracket^I$, because they may not be compatible with the time-derivatives for the differential equation, e.g. in $x' := 1; x' = 2$, with a x' mismatch. The final values $\omega(x')$ will coincide with the derivatives, though.

Functions and predicates are interpreted by I and are only influenced indirectly by ν through the values of their arguments. So $p(e) \rightarrow [x := x + 1]p(e)$ is valid if x is not in e since the change in x does not change whether $p(e)$ is true (Lemma 2). By contrast $p(x) \rightarrow [x := x + 1]p(x)$ is invalid, since it is false when $I(p) = \{d : d \leq 5\}$ and $\nu(x) = 4.5$. If the semantics of p were to depend on the state ν , then there would be no discernible relationship between the truth-values of p in different states, so not even $p \rightarrow [x := x + 1]p$ would be valid.

2.3 Static Semantics

The static semantics of \mathbf{dL} and HPs defines some aspects of their behavior that can be read off directly from their syntactic structure without running their programs or evaluating their dynamical effects. The most important aspects of the static semantics concern free or bound occurrences of variables (which are closely related to the notions of scope and definitions/uses in compilers). Bound variables x are those that are bound by $\forall x$ or $\exists x$, but also those that are bound by modalities such as $[x := 5y]$ or $\langle x' = 1 \rangle$ or $[x := 1 \cup x' = 1]$ or $[x := 1 \cup ?true]$.

The notions of free and bound variables are defined by simultaneous induction in the subsequent definitions: free variables for terms ($\mathbf{FV}(\theta)$), formulas ($\mathbf{FV}(\phi)$), and HPs ($\mathbf{FV}(\alpha)$), as well as bound variables for formulas ($\mathbf{BV}(\phi)$) and for HPs ($\mathbf{BV}(\alpha)$). For HPs, there will be a need to distinguish must-bound variables ($\mathbf{MBV}(\alpha)$) that are bound/written to on all executions of α from (may-)bound variables ($\mathbf{BV}(\alpha)$) which are bound on some (not necessarily all) execution paths of α , such as in $[x := 1 \cup (x := 0; y := x + 1)]$, which has bound variables $\{x, y\}$ but must-bound variables only $\{x\}$, because y is not written to in the first choice.

Definition 7 (Bound variable). The set $\mathbf{BV}(\phi) \subseteq \mathcal{V} \cup \mathcal{V}'$ of *bound variables* of \mathbf{dL} formula ϕ is defined inductively as

$$\begin{aligned} \mathbf{BV}(\theta \geq \eta) &= \mathbf{BV}(p(\theta_1, \dots, \theta_k)) = \emptyset \\ \mathbf{BV}(C(\phi)) &= \mathcal{V} \cup \mathcal{V}' \\ \mathbf{BV}(\neg\phi) &= \mathbf{BV}(\phi) \\ \mathbf{BV}(\phi \wedge \psi) &= \mathbf{BV}(\phi) \cup \mathbf{BV}(\psi) \\ \mathbf{BV}(\forall x \phi) &= \mathbf{BV}(\exists x \phi) = \{x\} \cup \mathbf{BV}(\phi) \\ \mathbf{BV}([\alpha]\phi) &= \mathbf{BV}(\langle \alpha \rangle \phi) = \mathbf{BV}(\alpha) \cup \mathbf{BV}(\phi) \end{aligned}$$

Definition 8 (Free variable). The set $\mathbf{FV}(\theta) \subseteq \mathcal{V} \cup \mathcal{V}'$ of *free variables* of term θ , i.e. those that occur in θ , is defined inductively as

$$\begin{aligned} \mathbf{FV}(x) &= \{x\} \\ \mathbf{FV}(x') &= \{x'\} \\ \mathbf{FV}(f(\theta_1, \dots, \theta_k)) &= \mathbf{FV}(\theta_1) \cup \dots \cup \mathbf{FV}(\theta_k) \\ \mathbf{FV}(\theta + \eta) &= \mathbf{FV}(\theta \cdot \eta) = \mathbf{FV}(\theta) \cup \mathbf{FV}(\eta) \\ \mathbf{FV}((\theta)') &= \mathbf{FV}(\theta) \cup \mathbf{FV}(\theta)' \end{aligned}$$

The set $\mathbf{FV}(\phi)$ of *free variables* of \mathbf{dL} formula ϕ , i.e. all those that occur in ϕ outside the scope of quantifiers or modalities binding it, is defined inductively as

$$\begin{aligned} \mathbf{FV}(\theta \geq \eta) &= \mathbf{FV}(\theta) \cup \mathbf{FV}(\eta) \\ \mathbf{FV}(p(\theta_1, \dots, \theta_k)) &= \mathbf{FV}(\theta_1) \cup \dots \cup \mathbf{FV}(\theta_k) \\ \mathbf{FV}(C(\phi)) &= \mathcal{V} \cup \mathcal{V}' \\ \mathbf{FV}(\neg\phi) &= \mathbf{FV}(\phi) \end{aligned}$$

$$\begin{aligned}
\text{FV}(\phi \wedge \psi) &= \text{FV}(\phi) \cup \text{FV}(\psi) \\
\text{FV}(\forall x \phi) &= \text{FV}(\exists x \phi) = \text{FV}(\phi) \setminus \{x\} \\
\text{FV}([\alpha]\phi) &= \text{FV}(\langle \alpha \rangle \phi) = \text{FV}(\alpha) \cup (\text{FV}(\phi) \setminus \text{MBV}(\alpha))
\end{aligned}$$

Soundness requires that $\text{FV}([\alpha]\phi)$ is not defined as $\text{FV}(\alpha) \cup (\text{FV}(\phi) \setminus \text{BV}(\alpha))$, otherwise $[x := 1 \cup y := 2]x \geq 1$ would have no free variables, but its truth-value depends on the initial value of x , demanding $\text{FV}([x := 1 \cup y := 2]x \geq 1) = \{x\}$. The simpler definition $\text{FV}([\alpha]\phi) = \text{FV}(\alpha) \cup \text{FV}(\phi)$ would be correct, but the results would be less precise then. Likewise for $\langle \alpha \rangle \phi$. Soundness requires $\text{FV}((\theta)')$ not to be defined as $\text{FV}(\theta)'$, because the value of $(xy)'$ depends on $\{x, x', y, y'\}$, since $(xy)'$ equals $x'y + xy'$ (Lemma 13).

The static semantics defines which variables are free so may be read ($\text{FV}(\alpha)$), which are bound ($\text{BV}(\alpha)$) so may be written to somewhere in α , and which are must-bound ($\text{MBV}(\alpha)$) so must be written to on all execution paths of α .

Definition 9 (Bound variable). The set $\text{BV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ of *bound variables* of HP α , i.e. all those that may potentially be written to, is defined inductively:

$$\begin{aligned}
\text{BV}(a) &= \mathcal{V} \cup \mathcal{V}' && \text{for program constant } a \\
\text{BV}(x := \theta) &= \{x\} \\
\text{BV}(x' := \theta) &= \{x'\} \\
\text{BV}(\psi) &= \emptyset \\
\text{BV}(x' = \theta \ \& \ \psi) &= \{x, x'\} \\
\text{BV}(\alpha \cup \beta) &= \text{BV}(\alpha; \beta) = \text{BV}(\alpha) \cup \text{BV}(\beta) \\
\text{BV}(\alpha^*) &= \text{BV}(\alpha)
\end{aligned}$$

Definition 10 (Must-bound variable). The set $\text{MBV}(\alpha) \subseteq \text{BV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ of *must-bound variables* of HP α , i.e. all those that must be written to on all paths of α , is defined inductively as

$$\begin{aligned}
\text{MBV}(a) &= \emptyset && \text{for program constant } a \\
\text{MBV}(\alpha) &= \text{BV}(\alpha) && \text{for other atomic HPs } \alpha \\
\text{MBV}(\alpha \cup \beta) &= \text{MBV}(\alpha) \cap \text{MBV}(\beta) \\
\text{MBV}(\alpha; \beta) &= \text{MBV}(\alpha) \cup \text{MBV}(\beta) \\
\text{MBV}(\alpha^*) &= \emptyset
\end{aligned}$$

Obviously, $\text{MBV}(\alpha) \subseteq \text{BV}(\alpha)$. If α is only built by sequential compositions from atomic programs without program constants, then $\text{MBV}(\alpha) = \text{BV}(\alpha)$.

Definition 11 (Free variable). The set $\text{FV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ of *free variables* of HP α , i.e. all those that may potentially be read, is defined inductively as

$$\begin{aligned}
\text{FV}(a) &= \mathcal{V} \cup \mathcal{V}' && \text{for program constant } a \\
\text{FV}(x := \theta) &= \text{FV}(x' := \theta) = \text{FV}(\theta)
\end{aligned}$$

$$\begin{aligned}
\text{FV}(\psi) &= \text{FV}(\psi) \\
\text{FV}(x' = \theta \ \& \ \psi) &= \{x\} \cup \text{FV}(\theta) \cup \text{FV}(\psi) \\
\text{FV}(\alpha \cup \beta) &= \text{FV}(\alpha) \cup \text{FV}(\beta) \\
\text{FV}(\alpha; \beta) &= \text{FV}(\alpha) \cup (\text{FV}(\beta) \setminus \text{MBV}(\alpha)) \\
\text{FV}(\alpha^*) &= \text{FV}(\alpha)
\end{aligned}$$

The *variables* of HP α , whether free or bound, are $\text{V}(\alpha) = \text{FV}(\alpha) \cup \text{BV}(\alpha)$.

The simpler definition $\text{FV}(\alpha \cup \beta) = \text{FV}(\alpha) \cup \text{FV}(\beta)$ would be correct, but the results would be less precise then. Unlike x , the left-hand side x' of differential equations is not added to the free variables of $\text{FV}(x' = \theta \ \& \ \psi)$, because its behavior does not depend on the initial value of differential symbols x' , only the initial value of x . Free and bound variables are the set of all variables \mathcal{V} and differential symbols \mathcal{V}' for program constants a , because their effect depends on the interpretation I , so may read and write all $\text{FV}(a) = \text{BV}(a) = \mathcal{V} \cup \mathcal{V}'$ but not on all paths $\text{MBV}(a) = \emptyset$. Subsequent results about free and bound variables are, thus, vacuously true when program constants occur. Corresponding observations hold for quantifier symbols.

The static semantics defines which variables are readable or writable. There may not be any run of α in which a variable is read or written to. If $x \notin \text{FV}(\alpha)$, though, then α cannot read the value of x . If $x \notin \text{BV}(\alpha)$, it cannot write to x . Def. 11 is parsimonious. For example, x is not a free variable of the following program

$$(x := 1 \cup x := 2); z := x + y$$

because x is never actually read, since x must have been defined on *every* execution path of the first part before being read by the second part. No execution of the above program, thus, depends on the initial value of x , which is why it is not a free variable. This would have been different for the simpler definition

$$\text{FV}(\alpha; \beta) = \text{FV}(\alpha) \cup \text{FV}(\beta)$$

There is a limit to the precision with which any static analysis can determine which variables are really read or written [12]. The static semantics in Def. 11 will, e.g., call x a free variable of the following program even though no execution could read it, because they fail test *?false* when running the branch reading x :

$$z := 0; (?false; z := z + x)^*$$

The *signature*, i.e. set of function, predicate, quantifier symbols, and program constants in ϕ is denoted by $\Sigma(\phi)$ (accordingly for terms and programs). It is defined like $\text{FV}(\phi)$ except that all occurrences are free. Variables in $\mathcal{V} \cup \mathcal{V}'$ are interpreted by state ν . The symbols in $\Sigma(\phi)$ are interpreted by interpretation I .

2.4 Correctness of Static Semantics

The following result reflects that HPs have bounded effect: for a variable x to be modified during a run of α , x needs to be a bound variable in HP α , i.e. $x \in \text{BV}(\alpha)$, so that α can write to x . The

converse is not true, because α may bind a variable x , e.g. by having an assignment to x , that never actually changes the value of x , such as $x := x$ or because the assignment can never be executed. The following program, for example, binds x but will never change the value of x because there is no way of satisfying the test $?false$: $(?false; x := 42) \cup z := x + 1$.

Lemma 1 (Bound effect lemma). *If $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, then $\nu = \omega$ on $BV(\alpha)^{\mathbb{C}}$.*

Proof. The proof is by a straightforward structural induction on α .

- Since $BV(a) = \mathcal{V} \cup \mathcal{V}'$, the statement is vacuously true for program constant a , because $BV(a)^{\mathbb{C}} = \emptyset$.
- $(\nu, \omega) \in \llbracket x := \theta \rrbracket^I = \{(\nu, \omega) : \omega = \nu \text{ except that } \llbracket x \rrbracket^I \omega = \llbracket \theta \rrbracket^I \nu\}$ implies that $\nu = \omega$ except for $\{x\} = BV(x := \theta)$.
- $(\nu, \omega) \in \llbracket x' := \theta \rrbracket^I = \{(\nu, \omega) : \omega = \nu \text{ except that } \llbracket x' \rrbracket^I \omega = \llbracket \theta \rrbracket^I \nu\}$ implies that $\nu = \omega$ except for $\{x'\} = BV(x' := \theta)$.
- $(\nu, \nu) \in \llbracket ?\psi \rrbracket^I = \{(\nu, \nu) : \nu \in \llbracket \psi \rrbracket^I \text{ i.e. } \nu \in \llbracket \psi \rrbracket^I\}$ fits to $BV(?\psi) = \emptyset$
- $(\nu, \omega) \in \llbracket x' = \theta \& \psi \rrbracket^I$ implies that $\nu = \omega$ except for the variables with differential equations, which are $\{x, x'\} = BV(x' = \theta \& \psi)$ observing that $\nu(x')$ and $\omega(x')$ may differ by Def. 6.
- $(\nu, \omega) \in \llbracket \alpha \cup \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \cup \llbracket \beta \rrbracket^I$ implies $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$ or $(\nu, \omega) \in \llbracket \beta \rrbracket^I$, which, by induction hypothesis, implies $\nu = \omega$ on $BV(\alpha)^{\mathbb{C}}$ or $\nu = \omega$ on $BV(\beta)^{\mathbb{C}}$, respectively. Either case implies $\nu = \omega$ on $(BV(\alpha) \cup BV(\beta))^{\mathbb{C}} = BV(\alpha \cup \beta)^{\mathbb{C}}$.
- $(\nu, \omega) \in \llbracket \alpha; \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \circ \llbracket \beta \rrbracket^I$, i.e. there is a μ such that $(\nu, \mu) \in \llbracket \alpha \rrbracket^I$ and $(\mu, \omega) \in \llbracket \beta \rrbracket^I$. So, by induction hypothesis, $\nu = \mu$ on $BV(\alpha)^{\mathbb{C}}$ and $\mu = \omega$ on $BV(\beta)^{\mathbb{C}}$. By transitivity, $\nu = \omega$ on $(BV(\alpha) \cup BV(\beta))^{\mathbb{C}} = BV(\alpha; \beta)^{\mathbb{C}}$.
- $(\nu, \omega) \in \llbracket \alpha^* \rrbracket^I = \bigcup_{n \in \mathbb{N}} \llbracket \alpha^n \rrbracket^I$, then there is an $n \in \mathbb{N}$ and a sequence $\nu_0 = \nu, \nu_1, \dots, \nu_n = \omega$ such that $(\nu_i, \nu_{i+1}) \in \llbracket \alpha \rrbracket^I$ for all $i < n$. By n uses of the induction hypothesis, $\nu_i = \nu_{i+1}$ on $BV(\alpha)^{\mathbb{C}}$ for all $i < n$. Thus, $\nu = \nu_0 = \nu_n = \omega$ on $BV(\alpha)^{\mathbb{C}} = BV(\alpha^*)^{\mathbb{C}}$.

□

Similarly, only $BV(\phi)$ change their value during the evaluation of formulas.

The value of a term only depends on the values of its free variables. When evaluating a term θ in two states $\nu, \tilde{\nu}$ that differ widely but agree on the free variables $FV(\theta)$ of θ , the values of θ in both states coincide. Accordingly for different interpretations I, J that agree on the symbols $\Sigma(\theta)$ that occur in θ .

Lemma 2 (Coincidence lemma). *If $\nu = \tilde{\nu}$ on $FV(\theta)$ and $I = J$ on $\Sigma(\theta)$, then $\llbracket \theta \rrbracket^I \nu = \llbracket \theta \rrbracket^J \tilde{\nu}$.*

Proof. The proof is by structural induction on θ .

- $\llbracket x \rrbracket^I \nu = \nu(x) = \tilde{\nu}(x) = \llbracket x \rrbracket^J \tilde{\nu}$ for variable x since $\nu = \tilde{\nu}$ on $\text{FV}(x) = \{x\}$.
- $\llbracket x' \rrbracket^I \nu = \nu(x') = \tilde{\nu}(x') = \llbracket x' \rrbracket^J \tilde{\nu}$ for differential symbol x' since $\nu = \tilde{\nu}$ on $\text{FV}(x') = \{x'\}$.
- $\llbracket f(\theta_1, \dots, \theta_k) \rrbracket^I \nu = I(f)(\llbracket \theta_1 \rrbracket^I \nu, \dots, \llbracket \theta_k \rrbracket^I \nu) \stackrel{\text{IH}}{=} J(f)(\llbracket \theta_1 \rrbracket^J \tilde{\nu}, \dots, \llbracket \theta_k \rrbracket^J \tilde{\nu}) = \llbracket f(\theta_1, \dots, \theta_k) \rrbracket^J \tilde{\nu}$ by induction hypothesis, because $\text{FV}(\theta_i) \subseteq \text{FV}(f(\theta_1, \dots, \theta_k))$ and I and J were assumed to agree on the function symbol f that occurs in the term.
- $\llbracket \theta + \eta \rrbracket^I \nu = \llbracket \theta \rrbracket^I \nu + \llbracket \eta \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \theta \rrbracket^J \tilde{\nu} + \llbracket \eta \rrbracket^J \tilde{\nu} = \llbracket \theta + \eta \rrbracket^J \tilde{\nu}$ by induction hypothesis, because $\text{FV}(\theta) \subseteq \text{FV}(\theta + \eta)$ and $\text{FV}(\eta) \subseteq \text{FV}(\theta + \eta)$.
- $\llbracket \theta \cdot \eta \rrbracket^I \nu = \llbracket \theta \rrbracket^I \nu \cdot \llbracket \eta \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \theta \rrbracket^J \tilde{\nu} \cdot \llbracket \eta \rrbracket^J \tilde{\nu} = \llbracket \theta \cdot \eta \rrbracket^J \tilde{\nu}$ by induction hypothesis, because $\text{FV}(\theta) \subseteq \text{FV}(\theta \cdot \eta)$ and $\text{FV}(\eta) \subseteq \text{FV}(\theta \cdot \eta)$.
- $\llbracket (\theta)' \rrbracket^I \nu = \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} = \sum_x \tilde{\nu}(x') \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} \stackrel{\text{IH}}{=} \sum_x \tilde{\nu}(x') \frac{\partial \llbracket \theta \rrbracket^J \tilde{\nu}_x^X}{\partial X}$ since $\nu = \tilde{\nu}$ on $\text{FV}((\theta)'),$ which includes all differential symbols x' for all $x \in \text{FV}(\theta)$ (the others have partial derivative 0 so do not contribute to the sum), and by induction hypothesis on the simpler term θ , because $\text{FV}(\theta) \subseteq \text{FV}((\theta)').$ Note that partial derivatives are functional, i.e. the partial derivatives by X of $\llbracket \theta \rrbracket^I \nu_x^X$ and $\llbracket \theta \rrbracket^J \tilde{\nu}_x^X$ agree since, by induction hypothesis, $\llbracket \theta \rrbracket^I \nu_x^X = \llbracket \theta \rrbracket^J \tilde{\nu}_x^X$ for all X since $\nu_x^X = \tilde{\nu}_x^X$ on $\{x\} \cup \text{FV}(\theta)$ since x is interpreted to be X in both states and $\nu = \tilde{\nu}$ on $\text{FV}(\theta)$ already.

□

By a more subtle argument, the values of $\text{d}\mathcal{L}$ formulas also only depend on the values of their free variables. When evaluating $\text{d}\mathcal{L}$ formula ϕ in two states $\nu, \tilde{\nu}$ that differ but agree on the free variables $\text{FV}(\phi)$ of ϕ , the (truth) values of ϕ in both states coincide. Lemma 3 and 4 are proved by simultaneous induction.

Lemma 3 (Coincidence lemma). *If $\nu = \tilde{\nu}$ on $\text{FV}(\phi)$ and $I = J$ on $\Sigma(\phi)$, then $\nu \in \llbracket \phi \rrbracket^I$ iff $\tilde{\nu} \in \llbracket \phi \rrbracket^J$.*

Proof. The proof is by structural induction on ϕ .

1. $\nu \in \llbracket p(\theta_1, \dots, \theta_k) \rrbracket^I$ iff $(\llbracket \theta_1 \rrbracket^I \nu, \dots, \llbracket \theta_k \rrbracket^I \nu) \in I(p)$ iff $(\llbracket \theta_1 \rrbracket^J \tilde{\nu}, \dots, \llbracket \theta_k \rrbracket^J \tilde{\nu}) \in J(p)$ iff $\tilde{\nu} \in \llbracket p(\theta_1, \dots, \theta_k) \rrbracket^J$ by Lemma 2 since $\text{FV}(\theta_i) \subseteq \text{FV}(p(\theta_1, \dots, \theta_k))$ and I and J were assumed to agree on the function symbol p that occurs in the formula.
2. $\nu \in \llbracket \theta \geq \eta \rrbracket^I$ iff $\llbracket \theta \rrbracket^I \nu \geq \llbracket \eta \rrbracket^I \nu$ iff $\llbracket \theta \rrbracket^J \tilde{\nu} \geq \llbracket \eta \rrbracket^J \tilde{\nu}$ iff $\tilde{\nu} \in \llbracket \theta \geq \eta \rrbracket^J$ by Lemma 2 since $\text{FV}(\theta) \cup \text{FV}(\eta) \subseteq \text{FV}(\theta \geq \eta)$ and the interpretation of \geq is fixed.
3. $\nu \in \llbracket C(\phi) \rrbracket^I = I(C)(\llbracket \phi \rrbracket^I)$ iff (IH) $\tilde{\nu} \in \llbracket C(\phi) \rrbracket^J = J(C)(\llbracket \phi \rrbracket^J)$ since $\nu = \tilde{\nu}$ on $\text{FV}(C(\phi)) = \mathcal{V} \cup \mathcal{V}'$, so $\nu = \tilde{\nu}$, and $I = J$ on $\Sigma(C(\phi)) = \{C\} \cup \Sigma(\phi)$, so $I(C) = J(C)$ and, by induction hypothesis, $\llbracket \phi \rrbracket^I = \llbracket \phi \rrbracket^J$.

4. $\nu \in \llbracket \neg\phi \rrbracket^I$ iff $\nu \notin \llbracket \phi \rrbracket^I$ iff (IH) $\tilde{\nu} \notin \llbracket \phi \rrbracket^J$ iff $\tilde{\nu} \in \llbracket \neg\phi \rrbracket^J$ by induction hypothesis as $\text{FV}(\neg\phi) = \text{FV}(\phi)$.
5. $\nu \in \llbracket \phi \wedge \psi \rrbracket^I$ iff $\nu \in \llbracket \phi \rrbracket^I \cap \llbracket \psi \rrbracket^I$ iff (IH) $\tilde{\nu} \in \llbracket \phi \rrbracket^J \cap \llbracket \psi \rrbracket^J$ iff $\tilde{\nu} \in \llbracket \phi \wedge \psi \rrbracket^J$ by induction hypothesis using $\text{FV}(\phi \wedge \psi) = \text{FV}(\phi) \cup \text{FV}(\psi)$.
6. $\nu \in \llbracket \exists x \phi \rrbracket^I$ iff $\nu_x^r \in \llbracket \phi \rrbracket^I$ for some $r \in \mathbb{R}$ iff $\tilde{\nu}_x^r \in \llbracket \phi \rrbracket^I$ for some $r \in \mathbb{R}$ iff (H) $\tilde{\nu} \in \llbracket \exists x \phi \rrbracket^J$ for the same r by induction hypothesis using that $\nu_x^r = \tilde{\nu}_x^r$ on $\text{FV}(\phi) \subseteq \{x\} \cup \text{FV}(\exists x \phi)$.
7. The case $\forall x \phi$ follows from the equivalence $\forall x \phi \equiv \neg \exists x \neg \phi$ using $\text{FV}(\neg \exists x \neg \phi) = \text{FV}(\forall x \phi)$.
8. $\nu \in \llbracket \langle \alpha \rangle \phi \rrbracket^I$ iff there is a ω such that $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$ and $\omega \in \llbracket \phi \rrbracket^I$. Since $\nu = \tilde{\nu}$ on $\text{FV}(\langle \alpha \rangle \phi) \supseteq \text{FV}(\alpha)$ and $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, Lemma 4 implies with $I = J$ on $\Sigma(\alpha)$ that there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$ and $\omega = \tilde{\omega}$ on $\text{FV}(\langle \alpha \rangle \phi) \cup \text{MBV}(\alpha) = \text{FV}(\alpha) \cup (\text{FV}(\phi) \setminus \text{MBV}(\alpha)) \cup \text{MBV}(\alpha) = \text{FV}(\alpha) \cup \text{FV}(\phi) \cup \text{MBV}(\alpha) \supseteq \text{FV}(\phi)$.

$$\begin{array}{ccc}
\nu & \xrightarrow{\alpha} & \omega \\
\text{on } \text{FV}(\langle \alpha \rangle \phi) \supseteq \text{FV}(\alpha) \Big\| & & \Big\| \text{on } \text{FV}(\langle \alpha \rangle \phi) \cup \text{MBV}(\alpha) \supseteq \text{FV}(\phi) \\
\tilde{\nu} & \xrightarrow[\exists]{\alpha} & \tilde{\omega}
\end{array}$$

Since, $\omega = \tilde{\omega}$ on $\text{FV}(\phi)$ and $I = J$ on $\Sigma(\phi)$, the induction hypothesis implies that $\tilde{\omega} \in \llbracket \phi \rrbracket^J$ since $\omega \in \llbracket \phi \rrbracket^I$. Since $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$, this implies $\tilde{\nu} \in \llbracket \langle \alpha \rangle \phi \rrbracket^J$.

9. $\nu \in \llbracket [\alpha] \phi \rrbracket^I = \llbracket \neg \langle \alpha \rangle \neg \phi \rrbracket^I$ iff $\nu \notin \llbracket \langle \alpha \rangle \neg \phi \rrbracket^I$ iff $\tilde{\nu} \notin \llbracket \langle \alpha \rangle \neg \phi \rrbracket^J$ iff $\tilde{\nu} \in \llbracket [\alpha] \phi \rrbracket^J$ by induction hypothesis using $\text{FV}(\langle \alpha \rangle \neg \phi) = \text{FV}([\alpha] \phi)$.

□

In a sense, the runs of an HP α also only depend on the values of its free variables, because its behavior cannot depend on the values of variables that it never reads. That is, if $\nu = \tilde{\nu}$ on $\text{FV}(\alpha)$ and $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, then there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$ and ω and $\tilde{\omega}$ agree in some sense. There is a subtlety, though. The resulting states ω and $\tilde{\omega}$ will only continue to agree on $\text{FV}(\alpha)$ and the variables that are bound on the particular path that α took for the transition $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$. On variables z that are neither free (so the initial states ν and $\tilde{\nu}$ have not been assumed to coincide) nor bound on the particular path that α took, ω and $\tilde{\omega}$ may continue to disagree, because z has not been written to.

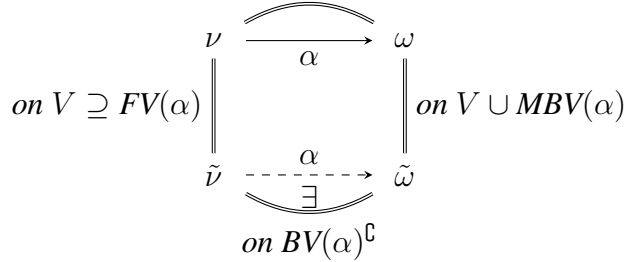
Example 1. Let $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$. It is not enough to assume $\nu = \tilde{\nu}$ only on $\text{FV}(\alpha)$ in order to guarantee $\omega = \tilde{\omega}$ on $\text{V}(\alpha)$ for some $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$, because

$$\alpha \stackrel{\text{def}}{=} x := 1 \cup y := 2$$

will force the final states to agree only on either x or on y , whichever one was assigned to during the respective run of α , not on both $\text{BV}(\alpha) = \{x, y\}$, even though any initial states $\nu, \tilde{\nu}$ agree on $\text{FV}(\alpha) = \emptyset$. Note that this can only happen because $\text{MBV}(\alpha) = \emptyset \neq \text{BV}(\alpha) = \{x, y\}$.

Yet, ω and $\tilde{\omega}$ agree on the variables that are bound on *all* paths of α , rather than somewhere in α . That is on the must-bound variables of α . If initial states agree on (at least) all free variables $FV(\alpha)$ that HP α may read, then the final states agree on those as well as on all variables that α must write, i.e. on $MBV(\alpha)$.

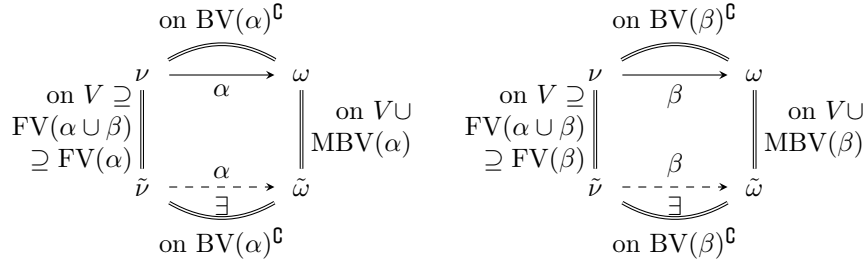
Lemma 4 (Coincidence lemma). *If $\nu = \tilde{\nu}$ on $V \supseteq FV(\alpha)$ and $I = J$ on $\Sigma(\alpha)$ and $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, then there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V \cup MBV(\alpha)$.*



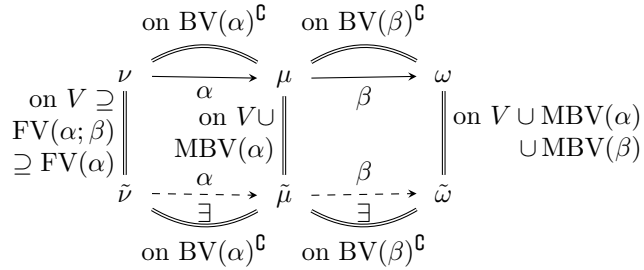
Proof. The proof is by induction on the structural complexity of α , where α^* is considered to be structurally more complex than HPs of any length but with less nested repetitions, which induces a well-founded order on HPs. For atomic programs α , for which $BV(\alpha) = MBV(\alpha)$, it is enough to conclude agreement on $V(\alpha) \stackrel{\text{def}}{=} FV(\alpha) \cup BV(\alpha) = FV(\alpha) \cup MBV(\alpha)$, because any variable in $V \setminus V(\alpha)$ is in $BV(\alpha)^G$, which remains unchanged by α according to Lemma 1.

- Since $FV(a) = \mathcal{V} \cup \mathcal{V}'$ so $\nu = \tilde{\nu}$, the statement is vacuously true for program constant a .
- $(\nu, \omega) \in \llbracket x := \theta \rrbracket^I = \{(\nu, \omega) : \omega = \nu \text{ except that } \llbracket x \rrbracket^I \omega = \llbracket \theta \rrbracket^I \nu\}$ then there is a transition $(\tilde{\nu}, \tilde{\omega}) \in \llbracket x := \theta \rrbracket^J$ and $\tilde{\omega}(x) = \llbracket x \rrbracket^J \tilde{\omega} = \llbracket \theta \rrbracket^J \tilde{\nu} = \llbracket \theta \rrbracket^I \nu = \llbracket x \rrbracket^I \omega = \nu(x)$ by Lemma 3, since $\nu = \tilde{\nu}$ on $FV(x := \theta) = FV(\theta)$ and $I = J$ on $\Sigma(\theta)$. So, $\omega = \tilde{\omega}$ on $BV(x := \theta) = \{x\}$. Also, $\nu = \omega$ on $BV(x := \theta)^G$ and $\tilde{\nu} = \tilde{\omega}$ on $BV(x := \theta)^G$ by Lemma 1. Since $\nu = \tilde{\nu}$ on $FV(x := \theta)$, these imply $\omega = \tilde{\omega}$ on $FV(x := \theta) \setminus BV(x := \theta)$. Since $\omega = \tilde{\omega}$ on $BV(x := \theta)$ had been shown already, this implies $\omega = \tilde{\omega}$ on $V(x := \theta)$.
- $(\nu, \omega) \in \llbracket x' := \theta \rrbracket^I = \{(\nu, \nu_{x'}^r) : r = \llbracket \theta \rrbracket^I \nu\}$. As $\llbracket \theta \rrbracket^I \nu = \llbracket \theta \rrbracket^J \tilde{\nu}$ by Lemma 2 since $FV(\theta) \subseteq FV(x' := \theta)$, this implies $(\tilde{\nu}, \tilde{\nu}_{x'}^r) \in \llbracket x' := \theta \rrbracket^J = \{(\tilde{\nu}, \tilde{\nu}_{x'}^r) : r = \llbracket \theta \rrbracket^J \tilde{\nu}\}$. By construction $\omega = \tilde{\nu}_{x'}^r$ on $BV(x' := \theta) = \{x'\}$ and they continue to agree on $FV(x' := \theta) \setminus BV(x' := \theta)$.
- $(\nu, \omega) \in \llbracket ?\psi \rrbracket^I = \{(\nu, \nu) : \nu \in \llbracket \psi \rrbracket^I \text{ i.e. } \nu \in \llbracket \psi \rrbracket^I\}$ then $\omega = \nu$ by Def. 6. Since, $\nu \in \llbracket \psi \rrbracket^I$ and $\nu = \tilde{\nu}$ on $FV(?\psi)$ and $I = J$ on $\Sigma(\psi)$, Lemma 3 implies that $\tilde{\nu} \in \llbracket \psi \rrbracket^J$, so $(\tilde{\nu}, \tilde{\nu}) \in \llbracket ?\psi \rrbracket^J$. So $\nu = \tilde{\nu}$ on $V(?\psi)$ since $BV(?\psi) = \emptyset$.
- $(\nu, \omega) \in \llbracket x' = \theta \& \psi \rrbracket^I$ implies that there is an $\tilde{\omega}$ reached from $\tilde{\nu}$ by following the differential equation for the same amount it took to reach ω from ν . The solution will be the same, because $I = J$ on $\Sigma(x' = \theta \& \psi)$ and $\nu = \tilde{\nu}$ on $FV(x' = \theta \& \psi)$, which, using Lemma 3, contains all the variables whose values the differential equation solution depends on. Thus, both solutions agree on all variables that evolve during the continuous evolution, i.e. $BV(x' = \theta \& \psi)$. Thus, $(\tilde{\nu}, \tilde{\omega}) \in \llbracket x' = \theta \& \psi \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V(x' = \theta \& \psi)$.

- $(\nu, \omega) \in \llbracket \alpha \cup \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \cup \llbracket \beta \rrbracket^I$ implies $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$ or $(\nu, \omega) \in \llbracket \beta \rrbracket^I$, which since $V \supseteq \text{FV}(\alpha \cup \beta) \supseteq \text{FV}(\alpha)$ and $V \supseteq \text{FV}(\alpha \cup \beta) \supseteq \text{FV}(\beta)$ implies, by induction hypothesis, that there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V \cup \text{MBV}(\alpha)$ or that there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \beta \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V \cup \text{MBV}(\beta)$, respectively. In either case, there is a $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \cup \beta \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V \cup \text{MBV}(\alpha \cup \beta)$, because $\llbracket \alpha \rrbracket^J \subseteq \llbracket \alpha \cup \beta \rrbracket^J$ and $\llbracket \beta \rrbracket^J \subseteq \llbracket \alpha \cup \beta \rrbracket^J$ and $\text{MBV}(\alpha \cup \beta) = \text{MBV}(\alpha) \cap \text{MBV}(\beta)$.



- $(\nu, \omega) \in \llbracket \alpha; \beta \rrbracket^I = \llbracket \alpha \rrbracket^I \circ \llbracket \beta \rrbracket^I$, i.e. there is a μ such that $(\nu, \mu) \in \llbracket \alpha \rrbracket^I$ and $(\mu, \omega) \in \llbracket \beta \rrbracket^I$. Since $V \supseteq \text{FV}(\alpha; \beta) \supseteq \text{FV}(\alpha)$, by induction hypothesis, there is a $\tilde{\mu}$ such that $(\tilde{\nu}, \tilde{\mu}) \in \llbracket \alpha \rrbracket^J$ and $\mu = \tilde{\mu}$ on $V \cup \text{MBV}(\alpha)$. Since $V \supseteq \text{FV}(\alpha; \beta)$, so $V \cup \text{MBV}(\alpha) \supseteq \text{FV}(\alpha; \beta) \cup \text{MBV}(\alpha) = \text{FV}(\alpha) \cup (\text{FV}(\beta) \setminus \text{MBV}(\alpha)) \cup \text{MBV}(\alpha) = \text{FV}(\alpha) \cup \text{FV}(\beta) \cup \text{MBV}(\alpha) \supseteq \text{FV}(\beta)$ by Def. 11, and since $(\mu, \omega) \in \llbracket \beta \rrbracket^I$, the induction hypothesis implies that there is an $\tilde{\omega}$ such that $(\tilde{\mu}, \tilde{\omega}) \in \llbracket \beta \rrbracket^J$ and $\omega = \tilde{\omega}$ on $(V \cup \text{MBV}(\alpha)) \cup \text{MBV}(\beta) = V \cup \text{MBV}(\alpha; \beta)$.



- $(\nu, \omega) \in \llbracket \alpha^* \rrbracket^I = \bigcup_{n \in \mathbb{N}} \llbracket \alpha^n \rrbracket^I$ iff there is an $n \in \mathbb{N}$ such that $(\nu, \omega) \in \llbracket \alpha^n \rrbracket^I$. The case $n = 0$ follows from the assumption $\nu = \tilde{\nu}$ on $V \supseteq \text{FV}(\alpha)$, since $\omega = \nu$ holds in that case and $\text{MBV}(\alpha^*) = \emptyset$. The case $n > 0$ proceeds as follows. Since $\text{FV}(\alpha^n) = \text{FV}(\alpha^*) = \text{FV}(\alpha)$, the induction hypothesis applied to the structurally simpler HP α^n implies that there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha^n \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V \cup \text{MBV}(\alpha^n) \supseteq V = V \cup \text{MBV}(\alpha^*)$, since $\text{MBV}(\alpha^*) = \emptyset$. Since $\llbracket \alpha^n \rrbracket^J \subseteq \llbracket \alpha^* \rrbracket^J$, this concludes the proof.

□

When assuming ν and $\tilde{\nu}$ to agree on all $V(\alpha)$, whether free or bound, ω and $\tilde{\omega}$ will continue to agree on $V(\alpha)$:

Corollary 5 (Coincidence lemma). *If $\nu = \tilde{\nu}$ on $V(\alpha)$ and $I = J$ on $\Sigma(\alpha)$ and $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, then there is an $\tilde{\omega}$ such that $(\tilde{\nu}, \tilde{\omega}) \in \llbracket \alpha \rrbracket^J$ and $\omega = \tilde{\omega}$ on $V(\alpha)$. The same continues to hold if $\nu = \tilde{\nu}$ only on $V(\alpha) \setminus \text{MBV}(\alpha)$.*

Proof. By Lemma 4 with $V = V(\alpha) \supseteq \text{FV}(\alpha)$ or $V = V(\alpha) \setminus \text{MBV}(\alpha)$, respectively. \square

Remark 1. Using hybrid computation sequences, the agreement in Lemma 4 continues to hold for $\omega = \tilde{\omega}$ on $V \cup W$, where W is the set of must-bound variables on the hybrid computation sequence that α actually took for the transition $(\nu, \omega) \in \llbracket \alpha \rrbracket^I$, which could be larger than $\text{MBV}(\alpha)$.

3 Uniform Substitutions

The uniform substitution rule US_1 from first-order logic [2, §35,40] substitutes *all* occurrences of predicate $p(\cdot)$ by a formula $\psi(\cdot)$, i.e. it replaces all occurrences of $p(\theta)$, for any (vectorial) term θ , by the corresponding $\psi(\theta)$ simultaneously:

$$(\text{US}_1) \frac{\phi}{\phi_{p(\cdot)}^{\psi(\cdot)}} \quad (\text{US}) \frac{\phi}{\sigma(\phi)}$$

Rule US_1 [9] requires all relevant substitutions of $\psi(\theta)$ for $p(\theta)$ to be admissible and requires that no $p(\theta)$ occurs in the scope of a quantifier or modality binding a variable of $\psi(\theta)$ other than the occurrences in θ ; see [2, §35,40].

This section considers a constructive definition of this proof rule that is more general: US . The $\text{d}\mathcal{L}$ calculus uses uniform substitutions that affect terms, formulas, and programs. A *uniform substitution* σ is a mapping from expressions of the form $f(\cdot)$ to terms $\sigma f(\cdot)$, from $p(\cdot)$ to formulas $\sigma p(\cdot)$, from $C(\cdot)$ to formulas $\sigma C(\cdot)$, and from program constants a to HPs σa . Vectorial extensions are accordingly for uniform substitutions of other arities $k \geq 0$. Here \cdot is a reserved function symbol of arity zero and $_$ a reserved quantifier symbol of arity zero. Figure 1 defines the result $\sigma(\phi)$ of applying to a $\text{d}\mathcal{L}$ formula ϕ the *uniform substitution* σ that uniformly replaces all occurrences of function f by a (instantiated) term and all occurrences of predicate p or quantifier C by a (instantiated) formula as well as of program constant a by a program. The notation $\sigma f(\cdot)$ denotes the replacement for $f(\cdot)$ according to σ , i.e. the value $\sigma f(\cdot)$ of function σ at $f(\cdot)$. By contrast, $\sigma(\phi)$ denotes the result of applying σ to ϕ according to Fig. 1 (likewise for $\sigma(\theta)$ and $\sigma(\alpha)$). The notation $f \in \sigma$ signifies that σ replaces f , i.e. $\sigma f(\cdot) \neq f(\cdot)$. Finally, σ is a total function when augmented with $\sigma g(\cdot) = g(\cdot)$ for all $g \notin \sigma$. Accordingly for predicate symbols, quantifiers, and program constants.

Definition 12 (Admissible uniform substitution). The uniform substitution σ is *U -admissible* for ϕ (or θ or α , respectively) with respect to the set $U \subseteq \mathcal{V} \cup \mathcal{V}'$ iff $\text{FV}(\sigma|_{\Sigma(\phi)}) \cap U = \emptyset$, where $\sigma|_{\Sigma(\phi)}$ is the restriction of σ that only replaces symbols that occur in ϕ and $\text{FV}(\sigma) = \bigcup_{f \in \sigma} \text{FV}(\sigma f(\cdot)) \cup \bigcup_{p \in \sigma} \text{FV}(\sigma p(\cdot))$ are the *free variables* that σ introduces. The uniform substitution σ is *admissible* for ϕ (or θ or α , respectively) iff all admissibility conditions during its application according to Fig. 1 hold, which check that the bound variables U of each operator are not free in the substitution

on its arguments, i.e. σ is U -admissible. Otherwise the substitution clashes so its result $\sigma(\phi)$ ($\sigma(\theta)$ or $\sigma(\alpha)$) is not defined.

US is only applicable if σ is admissible for ϕ . In all subsequent results, all applications of uniform substitutions are required to be defined (no clash).

$\sigma(x)$	$=$	x	for variable $x \in \mathcal{V}$
$\sigma(x')$	$=$	x'	for differential symbol $x' \in \mathcal{V}'$
$\sigma(f(\theta))$	$=$	$(\sigma(f))(\sigma(\theta)) \stackrel{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot))$	for function symbol $f \in \sigma$
$\sigma(g(\theta))$	$=$	$g(\sigma(\theta))$	for function symbol $g \notin \sigma$
$\sigma(\theta + \eta)$	$=$	$\sigma(\theta) + \sigma(\eta)$	
$\sigma(\theta \cdot \eta)$	$=$	$\sigma(\theta) \cdot \sigma(\eta)$	
$\sigma((\theta)')$	$=$	$(\sigma(\theta))'$	if $\sigma \mathcal{V} \cup \mathcal{V}'$ -admissible for θ
<hr/>			
$\sigma(\theta \geq \eta)$	\equiv	$\sigma(\theta) \geq \sigma(\eta)$	
$\sigma(p(\theta))$	\equiv	$(\sigma(p))(\sigma(\theta)) \stackrel{\text{def}}{=} \{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot))$	for predicate symbol $p \in \sigma$
$\sigma(q(\theta))$	\equiv	$q(\sigma(\theta))$	for predicate symbol $q \notin \sigma$
$\sigma(C(\phi))$	\equiv	$\sigma(C)(\sigma(\phi)) \stackrel{\text{def}}{=} \{_ \mapsto \sigma(\phi)\}(\sigma C(_))$	if $\sigma \mathcal{V} \cup \mathcal{V}'$ -admissible for ϕ , $C \in \sigma$
$\sigma(C(\phi))$	\equiv	$C(\sigma(\phi))$	if $\sigma \mathcal{V} \cup \mathcal{V}'$ -admissible for ϕ , $C \notin \sigma$
$\sigma(\neg\phi)$	\equiv	$\neg\sigma(\phi)$	
$\sigma(\phi \wedge \psi)$	\equiv	$\sigma(\phi) \wedge \sigma(\psi)$	
$\sigma(\forall x \phi)$	$=$	$\forall x \sigma(\phi)$	if $\sigma \{x\}$ -admissible for ϕ
$\sigma(\exists x \phi)$	$=$	$\exists x \sigma(\phi)$	if $\sigma \{x\}$ -admissible for ϕ
$\sigma([\alpha]\phi)$	$=$	$[\sigma(\alpha)]\sigma(\phi)$	if $\sigma \text{BV}(\sigma(\alpha))$ -admissible for ϕ
$\sigma(\langle\alpha\rangle\phi)$	$=$	$\langle\sigma(\alpha)\rangle\sigma(\phi)$	if $\sigma \text{BV}(\sigma(\alpha))$ -admissible for ϕ
<hr/>			
$\sigma(a)$	\equiv	σa	for program constant $a \in \sigma$
$\sigma(b)$	\equiv	b	for program constant $b \notin \sigma$
$\sigma(x := \theta)$	\equiv	$x := \sigma(\theta)$	
$\sigma(x' := \theta)$	\equiv	$x' := \sigma(\theta)$	
$\sigma(x' = \theta \ \& \ \psi)$	\equiv	$x' = \sigma(\theta) \ \& \ \sigma(\psi)$	if $\sigma \{x, x'\}$ -admissible for θ, ψ
$\sigma(? \psi)$	\equiv	$? \sigma(\psi)$	
$\sigma(\alpha \cup \beta)$	\equiv	$\sigma(\alpha) \cup \sigma(\beta)$	
$\sigma(\alpha; \beta)$	\equiv	$\sigma(\alpha); \sigma(\beta)$	if $\sigma \text{BV}(\sigma(\alpha))$ -admissible for β
$\sigma(\alpha^*)$	\equiv	$(\sigma(\alpha))^*$	if $\sigma \text{BV}(\sigma(\alpha))$ -admissible for α

Figure 1: Recursive application of uniform substitution σ

3.1 Correctness of Uniform Substitutions

Let I_p^R denote the interpretation that agrees with interpretation I except for the interpretation of predicate symbol p , which is changed to $R \subseteq \mathbb{R}$. Accordingly for predicate symbols of other arities, for function symbols f , and quantifiers C .

Corollary 6 (Substitution adjoints). *The adjoint interpretation $\sigma_\nu^* I$ to substitution σ for I, ν is the interpretation that agrees with I except that for each function symbol $f \in \sigma$, predicate symbol $p \in \sigma$, quantifier $C \in \sigma$, and program constant $a \in \sigma$:*

$$\begin{aligned}\sigma_\nu^* I(f) &: \mathbb{R} \rightarrow \mathbb{R}; d \mapsto \llbracket \sigma f(\cdot) \rrbracket^{I^d} \nu \\ \sigma_\nu^* I(p) &= \{d \in \mathbb{R} : \nu \in \llbracket \sigma p(\cdot) \rrbracket^{I^d}\} \\ \sigma_\nu^* I(C) &: \wp(\mathbb{R}) \rightarrow \wp(\mathbb{R}); R \mapsto \llbracket \sigma C(\cdot) \rrbracket^{I^R} \\ \sigma_\nu^* I(a) &= \llbracket \sigma a \rrbracket^I\end{aligned}$$

If $\nu = \omega$ on $FV(\sigma)$, then $\sigma_\nu^* I = \sigma_\omega^* I$. If σ is U -admissible for ϕ (or θ or α , respectively) and $\nu = \omega$ on U^\complement , then

$$\begin{aligned}\llbracket \theta \rrbracket^{\sigma_\nu^* I} &= \llbracket \theta \rrbracket^{\sigma_\omega^* I} \text{ i.e. } \llbracket \theta \rrbracket^{\sigma_\nu^* I} \mu = \llbracket \theta \rrbracket^{\sigma_\omega^* I} \mu \text{ for all } \mu \\ \llbracket \phi \rrbracket^{\sigma_\nu^* I} &= \llbracket \phi \rrbracket^{\sigma_\omega^* I} \\ \llbracket \alpha \rrbracket^{\sigma_\nu^* I} &= \llbracket \alpha \rrbracket^{\sigma_\omega^* I}\end{aligned}$$

Proof. For well-definedness of $\sigma_\nu^* I$, note that $\sigma_\nu^* I(f)$ is a smooth function since $\sigma f(\cdot)$ has smooth values. First, $\sigma_\nu^* I(a) = \llbracket \sigma a \rrbracket^I = \sigma_\omega^* I(a)$ holds because the adjoint to σ for I, ν in the case of programs is independent of ν (the program has access to its respective initial state at runtime). Likewise $\sigma_\nu^* I(C) = \sigma_\omega^* I(C)$ for quantifiers. By Lemma 2, $\llbracket \sigma f(\cdot) \rrbracket^{I^d} \nu = \llbracket \sigma f(\cdot) \rrbracket^{I^d} \omega$ when $\nu = \omega$ on $FV(\sigma f(\cdot))$. Also $\nu \in \llbracket \sigma p(\cdot) \rrbracket^{I^d}$ iff $\omega \in \llbracket \sigma p(\cdot) \rrbracket^{I^d}$ by Lemma 3 when $\nu = \omega$ on $FV(\sigma p(\cdot))$. Thus, $\sigma_\nu^* I = \sigma_\omega^* I$ when $\nu = \omega$ on $FV(\sigma)$.

If σ is U -admissible for ϕ (or θ or α), then $FV(\sigma f(\cdot)) \cap U = \emptyset$ so $FV(\sigma f(\cdot)) \subseteq U^\complement$ for every function symbol $f \in \Sigma(\phi)$ (or θ or α) and likewise for predicate symbols $p \in \Sigma(\phi)$. Since $\nu = \omega$ on U^\complement , so $\sigma_\omega^* I = \sigma_\nu^* I$ on the function and predicate symbols in $\Sigma(\phi)$ (or θ or α). Finally $\sigma_\omega^* I = \sigma_\nu^* I$ implies that $\omega \in \llbracket \phi \rrbracket^{\sigma_\omega^* I}$ iff $\nu \in \llbracket \phi \rrbracket^{\sigma_\nu^* I}$ by Lemma 3 and that $\llbracket \theta \rrbracket^{\sigma_\nu^* I} = \llbracket \theta \rrbracket^{\sigma_\omega^* I}$ by Lemma 2 and that $\llbracket \alpha \rrbracket^{\sigma_\omega^* I} = \llbracket \alpha \rrbracket^{\sigma_\nu^* I}$ by Lemma 4. \square

Substituting equals for equals is sound by the compositional semantics of \mathbf{dL} . The more general uniform substitutions are still sound, because interpretations of uniform substitutes correspond to interpretations of their adjoints. The semantic modification of adjoint interpretations has the same effect as the syntactic uniform substitution, proved by simultaneous induction. Recall that all substitutions in the following are assumed to be defined (not clash), otherwise the lemmas make no claim.

Lemma 7 (Uniform substitution lemma). *The uniform substitution σ and its adjoint interpretation $\sigma_\nu^* I, \nu$ to σ for I, ν have the same term semantics:*

$$\llbracket \sigma(\theta) \rrbracket^I \nu = \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu$$

Proof. The proof is by structural induction on θ .

- $\llbracket \sigma(x) \rrbracket^I \nu = \llbracket x \rrbracket^I \nu = \nu(x) = \llbracket x \rrbracket^{\sigma_\nu^* I} \nu$ since $x \notin \sigma$ for variable $x \in \mathcal{V}$

- $\llbracket \sigma(x') \rrbracket^I \nu = \llbracket x' \rrbracket^I \nu = \nu(x') = \llbracket x' \rrbracket^{\sigma_\nu^* I} \nu$ as $x' \notin \sigma$ for differential symbol $x' \in \mathcal{V}'$
- $\llbracket \sigma(f(\theta)) \rrbracket^I \nu = \llbracket (\sigma(f))(\sigma(\theta)) \rrbracket^I \nu = \llbracket \{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot)) \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \sigma f(\cdot) \rrbracket^{I^d} \nu = (\sigma_\nu^* I(f))(d) = (\sigma_\nu^* I(f))(\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu) = \llbracket f(\theta) \rrbracket^{\sigma_\nu^* I} \nu$ with $d \stackrel{\text{def}}{=} \llbracket \sigma(\theta) \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu$ by using the induction hypothesis twice, once for $\sigma(\theta)$ on the smaller θ and once for $\{\cdot \mapsto \sigma(\theta)\}(\sigma f(\cdot))$ on the possibly bigger term $\sigma f(\cdot)$ but the structurally simpler uniform substitution $\{\cdot \mapsto \sigma(\theta)\}(\dots)$ that is a substitution on the symbol \cdot of arity zero, not a substitution of functions with arguments. For well-foundedness of the induction note that the \cdot substitution only happens for function symbols f with at least one argument θ (for $f \in \sigma$).
- $\llbracket \sigma(g(\theta)) \rrbracket^I \nu = \llbracket g(\sigma(\theta)) \rrbracket^I \nu = I(g)(\llbracket \sigma(\theta) \rrbracket^I \nu) \stackrel{\text{IH}}{=} I(g)(\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu) = \sigma_\nu^* I(g)(\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu) = \llbracket g(\theta) \rrbracket^{\sigma_\nu^* I} \nu$ by induction hypothesis and since $I(g) = \sigma_\nu^* I(g)$ as the interpretation of g does not change in $\sigma_\nu^* I$ for $g \notin \sigma$.
- $\llbracket \sigma(\theta + \eta) \rrbracket^I \nu = \llbracket \sigma(\theta) + \sigma(\eta) \rrbracket^I \nu = \llbracket \sigma(\theta) \rrbracket^I \nu + \llbracket \sigma(\eta) \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu + \llbracket \eta \rrbracket^{\sigma_\nu^* I} \nu = \llbracket \theta + \eta \rrbracket^{\sigma_\nu^* I} \nu$ by induction hypothesis.
- $\llbracket \sigma(\theta \cdot \eta) \rrbracket^I \nu = \llbracket \sigma(\theta) \cdot \sigma(\eta) \rrbracket^I \nu = \llbracket \sigma(\theta) \rrbracket^I \nu \cdot \llbracket \sigma(\eta) \rrbracket^I \nu \stackrel{\text{IH}}{=} \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu \cdot \llbracket \eta \rrbracket^{\sigma_\nu^* I} \nu = \llbracket \theta \cdot \eta \rrbracket^{\sigma_\nu^* I} \nu$ by induction hypothesis.
- $\llbracket \sigma((\theta)') \rrbracket^I \nu = \llbracket (\sigma(\theta))' \rrbracket^I \nu = \sum_x \nu(x') \frac{\partial \llbracket \sigma(\theta) \rrbracket^I \nu_x^X}{\partial X} \stackrel{\text{IH}}{=} \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu_x^X}{\partial X} = \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu_x^X}{\partial X} = \llbracket (\theta)' \rrbracket^{\sigma_\nu^* I} \nu$ by induction hypothesis, provided σ is $\mathcal{V} \cup \mathcal{V}'$ -admissible for θ , i.e. does not introduce any variables or differential symbols, so that Corollary 6 implies $\sigma_\nu^* I = \sigma_\omega^* I$ for all ν, ω (that agree on $(\mathcal{V} \cup \mathcal{V}')^c = \emptyset$, which imposes no condition on ν, ω).

□

Lemma 8 (Uniform substitution lemma). *The uniform substitution σ and its adjoint interpretation $\sigma_\nu^* I, \nu$ to σ for I, ν have the same formula semantics:*

$$\nu \in \llbracket \sigma(\phi) \rrbracket^I \text{ iff } \nu \in \llbracket \phi \rrbracket^{\sigma_\nu^* I}$$

Proof. The proof is by structural induction on ϕ .

- $\nu \in \llbracket \sigma(\theta \geq \eta) \rrbracket^I \text{ iff } \nu \in \llbracket \sigma(\theta) \geq \sigma(\eta) \rrbracket^I \text{ iff } \llbracket \sigma(\theta) \rrbracket^I \nu \geq \llbracket \sigma(\eta) \rrbracket^I \nu$, by Lemma 7, iff $\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu \geq \llbracket \eta \rrbracket^{\sigma_\nu^* I} \nu$ iff $\llbracket \theta \geq \eta \rrbracket^{\sigma_\nu^* I} \nu$
- $\nu \in \llbracket \sigma(p(\theta)) \rrbracket^I \text{ iff } \nu \in \llbracket (\sigma(p))(\sigma(\theta)) \rrbracket^I \text{ iff } \nu \in \llbracket \{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot)) \rrbracket^I \text{ iff, by IH, } \nu \in \llbracket \sigma p(\cdot) \rrbracket^{I^d} \text{ iff } d \in \sigma_\nu^* I(p) \text{ iff } (\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu) \in \sigma_\nu^* I(p) \text{ iff } \nu \in \llbracket p(\theta) \rrbracket^{\sigma_\nu^* I} \text{ with } d \stackrel{\text{def}}{=} \llbracket \sigma(\theta) \rrbracket^I \nu = \llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu$ by using Lemma 7 for $\sigma(\theta)$ and by using the induction hypothesis for $\{\cdot \mapsto \sigma(\theta)\}(\sigma p(\cdot))$ on the possibly bigger formula $\sigma p(\cdot)$ but the structurally simpler uniform substitution $\{\cdot \mapsto \sigma(\theta)\}(\dots)$ that is a mere substitution on symbol \cdot of arity zero, not a substitution of predicates (for $p \in \sigma$).

- $\nu \in \llbracket \sigma(q(\theta)) \rrbracket^I$ iff $\nu \in \llbracket q(\sigma(\theta)) \rrbracket^I$ iff $(\llbracket \sigma(\theta) \rrbracket^I \nu) \in I(q)$ so, by Lemma 7, iff $(\llbracket \theta \rrbracket^{\sigma^* I} \nu) \in I(q)$ iff $(\llbracket \theta \rrbracket^{\sigma^* I} \nu) \in \sigma^* I(q)$ iff $\nu \in \llbracket q(\theta) \rrbracket^{\sigma^* I}$ since $I(q) = \sigma^* I(q)$ as the interpretation of q does not change in $\sigma^* I$ (for $q \notin \sigma$)
- For the case $\sigma(C(\phi))$, first show $\llbracket \sigma(\phi) \rrbracket^I = \llbracket \phi \rrbracket^{\sigma^* I}$. By induction hypothesis for the smaller ϕ : $\omega \in \llbracket \sigma(\phi) \rrbracket^I$ iff $\omega \in \llbracket \phi \rrbracket^{\sigma^* I}$, where $\llbracket \phi \rrbracket^{\sigma^* I} = \llbracket \phi \rrbracket^{\sigma^* I}$ by Corollary 6 for all ν, ω (that agree on $(\mathcal{V} \cup \mathcal{V}')^c = \emptyset$, which imposes no condition on ν, ω) since σ is $\mathcal{V} \cup \mathcal{V}'$ -admissible for ϕ . The proof then proceeds:
 $\nu \in \llbracket \sigma(C(\phi)) \rrbracket^I = \llbracket \sigma(C)(\sigma(\phi)) \rrbracket^I = \llbracket \{- \mapsto \sigma(\phi)\}(\sigma C(-)) \rrbracket^I$, so, by induction hypothesis for the structurally simpler uniform substitution $\{- \mapsto \sigma(\phi)\}$ that is a mere substitution on symbol $-$ of arity zero, iff $\nu \in \llbracket \sigma C(-) \rrbracket^{I^R}$ since the adjoint to $\{- \mapsto \sigma(\phi)\}$ is I_-^R with $R \stackrel{\text{def}}{=} \llbracket \sigma(\phi) \rrbracket^I$.
 Also $\nu \in \llbracket C(\phi) \rrbracket^{\sigma^* I} = \sigma^* I(C)(\llbracket \phi \rrbracket^{\sigma^* I}) = \llbracket \sigma C(-) \rrbracket^{I^R}$ for $R = \llbracket \phi \rrbracket^{\sigma^* I} = \llbracket \sigma(\phi) \rrbracket^I$ by induction hypothesis. Both sides are, thus, equivalent.
- The case $\sigma(C(\phi))$ for $C \notin \sigma$ again first shows $\llbracket \sigma(\phi) \rrbracket^I = \llbracket \phi \rrbracket^{\sigma^* I}$ for all ν using that σ is $\mathcal{V} \cup \mathcal{V}'$ -admissible for ϕ . Then $\nu \in \llbracket \sigma(C(\phi)) \rrbracket^I = \llbracket C(\sigma(\phi)) \rrbracket^I = I(C)(\llbracket \sigma(\phi) \rrbracket^I) = I(C)(\llbracket \phi \rrbracket^{\sigma^* I}) = \sigma^* I(C)(\llbracket \phi \rrbracket^{\sigma^* I}) = \llbracket C(\phi) \rrbracket^{\sigma^* I}$ iff $\nu \in \llbracket C(\phi) \rrbracket^{\sigma^* I}$
- $\nu \in \llbracket \sigma(\neg \phi) \rrbracket^I$ iff $\nu \in \llbracket \neg \sigma(\phi) \rrbracket^I$ iff $\nu \notin \llbracket \sigma(\phi) \rrbracket^I$, by induction hypothesis, iff $\nu \notin \llbracket \phi \rrbracket^{\sigma^* I}$ iff $\nu \in \llbracket \neg \phi \rrbracket^{\sigma^* I}$
- $\nu \in \llbracket \sigma(\phi \wedge \psi) \rrbracket^I$ iff $\nu \in \llbracket \sigma(\phi) \wedge \sigma(\psi) \rrbracket^I$ iff $\nu \in \llbracket \sigma(\phi) \rrbracket^I$ and $\nu \in \llbracket \sigma(\psi) \rrbracket^I$, by induction hypothesis, iff $\nu \in \llbracket \phi \rrbracket^{\sigma^* I}$ and $\nu \in \llbracket \psi \rrbracket^{\sigma^* I}$ iff $\nu \in \llbracket \phi \wedge \psi \rrbracket^{\sigma^* I}$
- $\nu \in \llbracket \sigma(\exists x \phi) \rrbracket^I$ iff $\nu \in \llbracket \exists x \sigma(\phi) \rrbracket^I$ (provided σ is $\{x\}$ -admissible for ϕ) iff $\nu_x^d \in \llbracket \sigma(\phi) \rrbracket^I$ for some d , so, by induction hypothesis, iff $\nu_x^d \in \llbracket \phi \rrbracket^{\sigma^* I}$ for some d , which is equivalent to $\nu_x^d \in \llbracket \phi \rrbracket^{\sigma^* I}$ by Corollary 6 as σ is $\{x\}$ -admissible for ϕ and $\nu = \nu_x^d$ on $\{x\}^c$. Thus, this is equivalent to $\nu \in \llbracket \exists x \phi \rrbracket^{\sigma^* I}$.
- The case $\nu \in \llbracket \sigma(\forall x \phi) \rrbracket^I$ follows by duality $\forall x \phi \equiv \neg \exists x \neg \phi$, which is respected in the definition of uniform substitutions.
- $\nu \in \llbracket \sigma(\langle \alpha \rangle \phi) \rrbracket^I$ iff $\nu \in \llbracket \langle \sigma(\alpha) \rangle \sigma(\phi) \rrbracket^I$ (provided σ is $\text{BV}(\sigma(\alpha))$ -admissible for ϕ) iff there is a ω such that $(\nu, \omega) \in \llbracket \sigma(\alpha) \rrbracket^I$ and $\omega \in \llbracket \sigma(\phi) \rrbracket^I$, which, by Lemma 9 and induction hypothesis, respectively, is equivalent to: there is a ω such that $(\nu, \omega) \in \llbracket \alpha \rrbracket^{\sigma^* I}$ and $\omega \in \llbracket \phi \rrbracket^{\sigma^* I}$, which is equivalent to $\nu \in \llbracket \langle \alpha \rangle \phi \rrbracket^{\sigma^* I}$, because $\omega \in \llbracket \phi \rrbracket^{\sigma^* I}$ is equivalent to $\omega \in \llbracket \phi \rrbracket^{\sigma^* I}$ by Corollary 6 as σ is $\text{BV}(\sigma(\alpha))$ -admissible for ϕ and $\nu = \omega$ on $\text{BV}(\sigma(\alpha))^c$ by Lemma 1 since $(\nu, \omega) \in \llbracket \sigma(\alpha) \rrbracket^I$.
- The case $\nu \in \llbracket \sigma([\alpha] \phi) \rrbracket^I$ follows by duality $[\alpha] \phi \equiv \neg \langle \alpha \rangle \neg \phi$, which is respected in the definition of uniform substitutions.

□

Lemma 9 (Uniform substitution lemma). *The uniform substitution σ and its adjoint interpretation $\sigma_\nu^* I$, ν to σ for I , ν have the same program semantics:*

$$(\nu, \omega) \in \llbracket \sigma(\alpha) \rrbracket^I \text{ iff } (\nu, \omega) \in \llbracket \alpha \rrbracket^{\sigma_\nu^* I}$$

Proof. The proof is by structural induction on α .

- $(\nu, \omega) \in \llbracket \sigma(a) \rrbracket^I = \llbracket \sigma a \rrbracket^I = \sigma_\nu^* I(a) = \llbracket a \rrbracket^{\sigma_\nu^* I}$ for program constant $a \in \sigma$ (the proof is accordingly for $a \notin \sigma$).
- $(\nu, \omega) \in \llbracket \sigma(x := \theta) \rrbracket^I = \llbracket x := \sigma(\theta) \rrbracket^I$ iff $\omega = \nu_x^{\llbracket \sigma(\theta) \rrbracket^I \nu} = \nu_x^{\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu}$ by Lemma 8, which is, thus, equivalent to $(\nu, \omega) \in \llbracket x := \theta \rrbracket^{\sigma_\nu^* I}$.
- $(\nu, \omega) \in \llbracket \sigma(x' := \theta) \rrbracket^I = \llbracket x' := \sigma(\theta) \rrbracket^I$ iff $\omega = \nu_{x'}^{\llbracket \sigma(\theta) \rrbracket^I \nu} = \nu_{x'}^{\llbracket \theta \rrbracket^{\sigma_\nu^* I} \nu}$ by Lemma 8, which is, thus, equivalent to $(\nu, \omega) \in \llbracket x' := \theta \rrbracket^{\sigma_\nu^* I}$.
- $(\nu, \omega) \in \llbracket \sigma(? \psi) \rrbracket^I = \llbracket ? \sigma(\psi) \rrbracket^I$ iff $\omega = \nu$ and $\nu \in \llbracket \sigma(\psi) \rrbracket^I$, iff, by Lemma 8, $\omega = \nu$ and $\nu \in \llbracket \psi \rrbracket^{\sigma_\nu^* I}$, which is equivalent to $(\nu, \omega) \in \llbracket ? \psi \rrbracket^{\sigma_\nu^* I}$.

- $(\nu, \omega) \in \llbracket \sigma(x' = \theta \& \psi) \rrbracket^I = \llbracket x' = \sigma(\theta) \& \sigma(\psi) \rrbracket^I$ (provided $\sigma \{x, x'\}$ -admissible for θ, ψ) iff $\exists \varphi : [0, T] \rightarrow \mathcal{S}$ with $\varphi(0) = \nu, \varphi(T) = \omega$ and for all $t \geq 0$: $\varphi'(t) = \llbracket \sigma(\theta) \rrbracket^I \varphi(t) = \llbracket \theta \rrbracket^{\sigma_{\varphi(t)}^* I} \varphi(t)$ by Lemma 7 as well as $\varphi(t) \in \llbracket \sigma(\psi) \rrbracket^I$, which, by Lemma 8, is equivalent to $\varphi(t) \in \llbracket \psi \rrbracket^{\sigma_{\varphi(t)}^* I}$.

Also $(\nu, \omega) \in \llbracket x' = \theta \& \psi \rrbracket^{\sigma_\nu^* I}$ iff $\exists \varphi : [0, T] \rightarrow \mathcal{S}$ with $\varphi(0) = \nu, \varphi(T) = \omega$ and for all $t \geq 0$: $\varphi'(t) = \llbracket \theta \rrbracket^{\sigma_\nu^* I} \varphi(t)$ and $\varphi(t) \in \llbracket \psi \rrbracket^{\sigma_\nu^* I}$. Finally, $\llbracket \theta \rrbracket^{\sigma_\nu^* I} = \llbracket \theta \rrbracket^{\sigma_{\varphi(t)}^* I}$ and $\llbracket \psi \rrbracket^{\sigma_{\varphi(t)}^* I} = \llbracket \psi \rrbracket^{\sigma_\nu^* I}$ by Corollary 6 since σ is $\{x, x'\}$ -admissible for θ, ψ and $\nu = \varphi(t)$ on $\text{BV}(x' = \theta \& \psi)^\mathbb{C} = \{x, x'\}^\mathbb{C}$ by Lemma 1.

- $(\nu, \omega) \in \llbracket \sigma(\alpha \cup \beta) \rrbracket^I = \llbracket \sigma(\alpha) \cup \sigma(\beta) \rrbracket^I = \llbracket \sigma(\alpha) \rrbracket^I \cup \llbracket \sigma(\beta) \rrbracket^I$, which, by induction hypothesis, is equivalent to $(\nu, \omega) \in \llbracket \alpha \rrbracket^{\sigma_\nu^* I}$ or $(\nu, \omega) \in \llbracket \beta \rrbracket^{\sigma_\nu^* I}$, which is equivalent to $(\nu, \omega) \in \llbracket \alpha \rrbracket^{\sigma_\nu^* I} \cup \llbracket \beta \rrbracket^{\sigma_\nu^* I} = \llbracket \alpha \cup \beta \rrbracket^{\sigma_\nu^* I}$.
- $(\nu, \omega) \in \llbracket \sigma(\alpha; \beta) \rrbracket^I = \llbracket \sigma(\alpha); \sigma(\beta) \rrbracket^I = \llbracket \sigma(\alpha) \rrbracket^I \circ \llbracket \sigma(\beta) \rrbracket^I$ (provided σ is $\text{BV}(\sigma(\alpha))$ -admissible for β) iff there is a μ such that $(\nu, \mu) \in \llbracket \sigma(\alpha) \rrbracket^I$ and $(\mu, \omega) \in \llbracket \sigma(\beta) \rrbracket^I$, which, by induction hypothesis, is equivalent to $(\nu, \mu) \in \llbracket \alpha \rrbracket^{\sigma_\nu^* I}$ and $(\mu, \omega) \in \llbracket \beta \rrbracket^{\sigma_\mu^* I}$. Yet, $\llbracket \beta \rrbracket^{\sigma_\mu^* I} = \llbracket \beta \rrbracket^{\sigma_\nu^* I}$ by Corollary 6, because σ is $\text{BV}(\sigma(\alpha))$ -admissible for β and $\nu = \omega$ on $\text{BV}(\sigma(\alpha))^\mathbb{C}$ by Lemma 1 since $(\nu, \mu) \in \llbracket \sigma(\alpha) \rrbracket^I$. Finally, $(\nu, \mu) \in \llbracket \alpha \rrbracket^{\sigma_\nu^* I}$ and $(\mu, \omega) \in \llbracket \beta \rrbracket^{\sigma_\nu^* I}$ for some μ is equivalent to $(\nu, \omega) \in \llbracket \alpha; \beta \rrbracket^{\sigma_\nu^* I}$.
- $(\nu, \omega) \in \llbracket \sigma(\alpha^*) \rrbracket^I = \llbracket (\sigma(\alpha))^* \rrbracket^I = (\llbracket \sigma(\alpha) \rrbracket^I)^* = \bigcup_{n \in \mathbb{N}} (\llbracket \sigma(\alpha) \rrbracket^I)^n$ (provided σ is $\text{BV}(\sigma(\alpha))$ -admissible for α) iff there are $n \in \mathbb{N}$ and $\nu_0 = \nu, \nu_1, \dots, \nu_n = \omega$ such that $(\nu_i, \nu_{i+1}) \in \llbracket \sigma(\alpha) \rrbracket^I$ for all $i < n$. By n uses of the induction hypothesis, this is equivalent to $(\nu_i, \nu_{i+1}) \in \llbracket \alpha \rrbracket^{\sigma_{\nu_i}^* I}$.

for all $i < n$, which is equivalent to $(\nu_i, \nu_{i+1}) \in \llbracket \alpha \rrbracket^{\sigma^* I}$ by Corollary 6 since σ is $\text{BV}(\sigma(\alpha))$ -admissible for α and $\nu_{i+1} = \nu_i$ on $\text{BV}(\sigma(\alpha))^c$ by Lemma 1 as $(\nu_i, \nu_{i+1}) \in \llbracket \sigma(\alpha) \rrbracket^I$ for all $i < n$. Thus, this is equivalent to $(\nu, \omega) \in \llbracket \alpha^* \rrbracket^{\sigma^* I} = (\llbracket \alpha \rrbracket^{\sigma^* I})^*$.

□

3.2 Soundness

The uniform substitution lemmas are the key insights for the soundness of US. US is only applicable if the uniform substitution is defined (does not clash).

Theorem 10 (Soundness of uniform substitution). *US is sound and so is its special case US_1 . That is, if their premise is valid, then so is their conclusion.*

Proof. Let the premise ϕ of US be valid, i.e. $\nu \in \llbracket \phi \rrbracket^I$ for all interpretations and states I, ν . To show that the conclusion is valid, consider any interpretation and state I, ν and show $\nu \in \llbracket \sigma(\phi) \rrbracket^I$. By Lemma 8, $\nu \in \llbracket \sigma(\phi) \rrbracket^I$ iff $\nu \in \llbracket \phi \rrbracket^{\sigma^* I}$. The latter holds, because $\nu \in \llbracket \phi \rrbracket^I$ for all I, ν , including for $\sigma^* I, \nu$, by premise. The rule US_1 is the special case of US where σ only substitutes predicate symbol p . □

4 Differential Dynamic Logic Axioms

Proof rules and axioms for a Hilbert-type axiomatization of $\text{d}\mathcal{L}$ from prior work [7] are shown in Fig. 2, except that, thanks to rule US, axioms and rules now comprise the finite list of $\text{d}\mathcal{L}$ formulas in Fig. 2 as opposed to an infinite collection of axioms from a finite list of axiom schemata along with schema variables, side conditions, and implicit instantiation rules. Soundness of the axioms in Fig. 2 follows from the soundness of corresponding axiom schemata [7], but would be easier to prove standalone, because it is a finite list of formulas without the need to prove soundness for all their instantiations. The rules in Fig. 2 are *axiomatic rules*, i.e. pairs of concrete formulas instantiated by US. Further, \bar{x} is the vector of all relevant variables, which is finite-dimensional, or, in practice, considered as a built-in vectorial term. Proofs in the uniform substitution $\text{d}\mathcal{L}$ calculus use US (and bound renaming such as $\forall x p(x) \leftrightarrow \forall y p(y)$) to instantiate the axioms from Fig. 2 to the required form. CT, CQ, CE are congruence rules, which are included for efficiency to use axioms in any context even if not needed for completeness.

Real Quantifiers. Besides (decidable) real arithmetic (whose use is denoted \mathbb{R}), complete axioms for first-order logic can be adopted to express universal instantiation $\forall i$ (if p is true of all x it is also true of constant symbol f), distributivity $\forall \rightarrow$, and vacuous quantification $\forall \vee$ (predicate p of arity zero does not depend on x).

$$(\forall i) \quad (\forall x p(x)) \rightarrow p(f)$$

$$(\forall \rightarrow) \quad \forall x (p(x) \rightarrow q(x)) \rightarrow (\forall x p(x) \rightarrow \forall x q(x))$$

$$(\forall \vee) \quad p \rightarrow \forall x p$$

$\langle \cdot \rangle$	$\langle a \rangle p(\bar{x}) \leftrightarrow \neg[a]\neg p(\bar{x})$	G	$\frac{p(\bar{x})}{[a]p(\bar{x})}$
$[:=]$	$[x := f]p(x) \leftrightarrow p(f)$	\forall	$\frac{p(x)}{\forall x p(x)}$
$[?]$	$[?q]p \leftrightarrow (q \rightarrow p)$	MP	$\frac{p \rightarrow q \quad p}{q}$
$[\cup]$	$[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})$	CT	$\frac{f(\bar{x}) = g(\bar{x})}{c(f(\bar{x})) = c(g(\bar{x}))}$
$[:,]$	$[a; b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x})$	CQ	$\frac{f(\bar{x}) = g(\bar{x})}{p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))}$
$[*]$	$[a^*]p(\bar{x}) \leftrightarrow p(\bar{x}) \wedge [a][a^*]p(\bar{x})$	CE	$\frac{p(\bar{x}) \leftrightarrow q(\bar{x})}{C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))}$
K	$[a](p(\bar{x}) \rightarrow q(\bar{x})) \rightarrow ([a]p(\bar{x}) \rightarrow [a]q(\bar{x}))$	US	$\frac{\phi}{\sigma(\phi)}$
I	$[a^*](p(\bar{x}) \rightarrow [a]p(\bar{x})) \rightarrow (p(\bar{x}) \rightarrow [a^*]p(\bar{x}))$		
V	$p \rightarrow [a]p$		

Figure 2: Differential dynamic logic axioms and proof rules

The Significance of Clashes. This section illustrates how soundness-critical it is for US to produce substitution clashes by showing unsound naïve proof attempts that US prevents successfully. US clashes for substitutions that introduce a free variable into a bound context. Even an occurrence of $p(x)$ in a context where x is bound does not allow mentioning x in the replacement except in the \bullet places:

$$\text{clash} \not\vdash \frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x + 1]x \neq x \leftrightarrow x + 1 \neq x} \quad \sigma = \{f \mapsto x + 1, p(\bullet) \mapsto (\bullet \neq x)\}$$

US can directly handle even nontrivial binding structures, though, e.g. from $[:=]$ with the substitution $\sigma = \{f \mapsto x^2, p(\bullet) \mapsto [(z := \bullet + z)^*; z := \bullet + yz]y \geq \bullet\}$:

$$\text{US} \frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(z := x + z)^*; z := x + yz]y \geq x \leftrightarrow [(z := x^2 + z)^*; z := x^2 + yz]y \geq x^2}$$

Similarly from $[:=]$ with $\{f \mapsto x^2, p(\bullet) \mapsto [(y := y + 1 \cup z := \bullet + z^*); z := \bullet + yz]y > \bullet\}$:

$$\text{US} \frac{[x := f]p(x) \leftrightarrow p(f)}{[x := x^2][(y := y + 1 \cup z := x + z^*); z := x + yz]y > x \leftrightarrow [(y := y + 1 \cup z := x^2 + z^*); z := x^2 + yz]y > x^2}$$

It is soundness-critical that US clashes when trying to instantiate p in V_\forall with a formula that mentions the bound variable x :

$$\text{clash} \not\vdash \frac{p \rightarrow \forall x p}{x \geq 0 \rightarrow \forall x x \geq 0} \quad \{p \mapsto x \geq 0\}$$

It is soundness-critical that US clashes when substituting p in vacuous program axiom V with a formula with a free occurrence of a variable bound by a :

$$\text{clash} \frac{p \rightarrow [a]p}{x \geq 0 \rightarrow [x := x - 1]x \geq 0} \quad \{a \mapsto x := x - 1, p \mapsto x \geq 0\}$$

Gödel's generalization rule G uses $p(\bar{x})$ instead of p from V, so allows the proof:

$$\text{US} \frac{(-x)^2 \geq 0}{[x := x - 1](-x)^2 \geq 0}$$

Let $\bar{x} = (x, y)$, $\{a \mapsto x := x + 1, b \mapsto x := 0; y := 0, p(\bar{x}) \mapsto x \geq y\}$, US derives:

$$\text{US} \frac{\begin{array}{c} * \\ [a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x}) \end{array}}{[x := x + 1 \cup (x := 0; y := 0)]x \geq y \leftrightarrow [x := x + 1]x \geq 0 \wedge [x := 0; y := 0]x \geq y}$$

With $\bar{x} = (x, y)$ and $\{a \mapsto x := x + 1 \cup y := 0, b \mapsto y := y + 1, p(\bar{x}) \mapsto x \geq y\}$ US derives:

$$\text{US} \frac{\begin{array}{c} * \\ [a; b]p(\bar{x}) \leftrightarrow [a][b]p(\bar{x}) \end{array}}{[(x := x + 1 \cup y := 0); y := y + 1]x \geq y \leftrightarrow [x := x + 1 \cup y := 0][y := y + 1]x \geq y}$$

Not all axioms fit to the uniform substitution framework. The Barcan axiom was used in a completeness proof for the Hilbert-type calculus for differential dynamic logic [7] (but not in the completeness proof for its sequent calculus [5]):

$$(B) \quad \forall x [\alpha]p(x) \rightarrow [\alpha]\forall x p(x) \quad (x \notin \alpha)$$

B is unsound without the restriction $x \notin \alpha$, though, so that the following would be an unsound axiom:

$$\forall x [a]p(x) \rightarrow [a]\forall x p(x) \quad (1)$$

because $x \notin a$ cannot be enforced for program constants, since their effect might very well depend on the value of x or since they might write to x . In (1), x cannot be written by a without violating soundness:

$$\frac{\forall x [a]p(x) \rightarrow [a]\forall x p(x)}{\forall x [x := 0]x \geq 0 \rightarrow [x := 0]\forall x x \geq 0} \quad \{a \mapsto x := 0, p(\cdot) \mapsto \cdot \geq 0\}$$

nor can x be read by a in (1) without violating soundness:

$$\frac{\forall x [a]p(x) \rightarrow [a]\forall x p(x)}{\forall x [?y = x^2]y = x^2 \rightarrow [?y = x^2]\forall x y = x^2} \quad \{a \mapsto ?y = x^2, p(\cdot) \mapsto y = \cdot^2\}$$

Thus, the completeness proof for differential dynamic logic from prior work [7] does not directly carry over. A more general completeness result for differential game logic [9] implies, however, that B is unnecessary for completeness.

5 Differential Equations and Differential Axioms

Section 4 leverages the first-order features of \mathbf{dL} and US to obtain a finite list of axioms without side-conditions. They lack axioms for differential equations, though. Classical calculi for \mathbf{dL} have axioms for replacing differential equations with a quantifier for time $t \geq 0$ and an assignment for their solutions $\bar{x}(t)$ [5, 7]. Besides being limited to simple differential equations, such axioms have the inherent side-condition “if $\bar{x}(t)$ is a solution of the differential equation $x' = \theta$ with symbolic initial value x ”. Such a side-condition is more difficult than occurrence and read/write conditions, but equally soundness-critical. This section leverages US and the new differential forms in \mathbf{dL} to obtain a logically internalized version of differential invariants and related proof rules for differential equations [6, 8] as axioms (without schema variables and free of side-conditions). These axioms can prove properties of more general “unsolvable” differential equations. They can also prove all properties of differential equations that can be proved with solutions [8] while guaranteeing correctness of the solution as part of the proof.

5.1 Differentials: Invariants, Cuts, Effects, and Ghosts

Figure 3 shows differential equation axioms for differential weakening (DW), differential cuts (DC), differential effect (DE), differential invariants (DI) [6], differential ghosts (DG) [8], solutions (DS), differential substitutions ($[\cdot :=]$), and differential axioms ($+$, \cdot , \circ'). Axioms identifying $(x)' = x'$ for variables $x \in \mathcal{V}$ and $(f)' = 0$ for functions f and number literals of arity 0 are used implicitly. Some axioms use reverse implications ($\phi \leftarrow \psi \equiv (\psi \rightarrow \phi)$) for emphasis.

$$\begin{aligned}
\text{DW} \quad & [x' = f(x) \ \& \ q(x)]q(x) \\
\text{DC} \quad & ([x' = f(x) \ \& \ q(x)]p(x) \leftrightarrow [x' = f(x) \ \& \ q(x) \ \wedge \ r(x)]p(x)) \leftarrow [x' = f(x) \ \& \ q(x)]r(x) \\
\text{DE} \quad & [x' = f(x) \ \& \ q(x)]p(x, x') \leftrightarrow [x' = f(x) \ \& \ q(x)][x' := f(x)]p(x, x') \\
\text{DI} \quad & [x' = f(x) \ \& \ q(x)]p(x) \leftarrow (q(x) \rightarrow p(x) \wedge [x' = f(x) \ \& \ q(x)](p(x))') \\
\text{DG} \quad & [x' = f(x) \ \& \ q(x)]p(x) \leftrightarrow \exists y [x' = f(x), y' = a(x)y + b(x) \ \& \ q(x)]p(x) \\
\text{DS} \quad & [x' = f \ \& \ q(x)]p(x) \leftrightarrow \forall t \geq 0 ((\forall 0 \leq s \leq t \ q(x + fs)) \rightarrow [x := x + ft]p(x)) \\
[\cdot :=] \quad & [x' := f]p(x') \leftrightarrow p(f) \\
+ \quad & (f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))' \\
\cdot \quad & (f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))' \\
\circ' \quad & [y := g(x)][y' := 1]((f(g(x)))' = (f(y))' \cdot (g(x))')
\end{aligned}$$

Figure 3: Differential equation axioms and differential axioms

The following proof proves a property of a differential equation using differential invariants without having to solve that differential equation. One use of US is shown explicitly, other uses of US are similar for DI, DE, $[':=]$ instances.

	$\frac{*}{(f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))'}$
US	$\frac{(x \cdot x)' = (x)' \cdot x + x \cdot (x)'}{(x \cdot x)' = x' \cdot x + x \cdot x'}$
CQ	$\frac{(xx)' \geq 0 \leftrightarrow x' \cdot x + xx' \geq 0}{(xx \geq 1)' \leftrightarrow x' \cdot x + xx' \geq 0}$
CE	$[x' = x^3][x' := x^3](xx \geq 1)'$
DE	$[x' = x^3](xx \geq 1)'$
DI	$xx \geq 1 \rightarrow [x' = x^3]xx \geq 1$

$\overline{[x' = f(x) \& q(x)](q(x) \rightarrow p(x))} \rightarrow [x' = f(x) \& q(x)]p(x)$ derives by K from DW. The converse $[x' = f(x) \& q(x)]p(x) \rightarrow [x' = f(x) \& q(x)](q(x) \rightarrow p(x))$ derives by K since G derives $[x' = f(x) \& q(x)](p(x) \rightarrow (q(x) \rightarrow p(x)))$.

$$\begin{array}{c}
\text{MP} \frac{\text{..} \rightarrow ((x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0)}{\text{use proof above}} \quad \frac{\text{US} \frac{\text{..} \rightarrow ((f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))')}{(x \cdot x)' = (x')' \cdot x + x \cdot (x')'}{(x \cdot x)' = x' \cdot x + x \cdot x'}}{\text{use proof above}} \\
\text{G} \frac{\text{K} \frac{\text{R} \frac{x^3 \cdot x + x \cdot x^3 \geq 0}{[x' := x^3]x' \cdot x + x \cdot x' \geq 0}}{[x' := x^3](x \cdot x)' \geq 0 \leftrightarrow [x' := x^3]x' \cdot x + x \cdot x' \geq 0}}{(x \cdot x)' \geq 0 \leftrightarrow x' \cdot x + x \cdot x' \geq 0} \\
\text{DI} \frac{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}
\end{array}$$

The proof uses (implicit) cuts with equivalences predicting the outcome of the right branch, which is simple but inconvenient. A constructive direct proof uses a free function symbol $j(x, x')$, instead, which is ultimately instantiated by US as in Theorem 14.

The same technique is helpful for invariant search, in which case a free predicate symbol $p(\bar{x})$ is used and instantiated by US lazily when the proof closes.

$$\begin{array}{c}
\text{R} \frac{\text{..} \rightarrow ((f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))')}{(x \cdot x)' = (x')' \cdot x + x \cdot (x')'}{\text{..} \rightarrow ((f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))')}{\text{..} \rightarrow ((f(\bar{x}) \cdot g(\bar{x}))' = (f(\bar{x}))' \cdot g(\bar{x}) + f(\bar{x}) \cdot (g(\bar{x}))')} \\
\text{G} \frac{\text{K} \frac{\text{R} \frac{x^3 \cdot x + x \cdot x^3 \geq 0}{j(x, x^3) \geq 0}}{[x' := x^3]j(x, x') \geq 0}}{[x' = x^3][x' := x^3]j(x, x') \geq 0} \\
\text{CE} \frac{\text{DE} \frac{\text{DI} \frac{x \cdot x \geq 1 \rightarrow [x' = x^3]x \cdot x \geq 1}{[x' = x^3](x \cdot x \geq 1)'}}{[x' = x^3][x' := x^3](x \cdot x \geq 1)'}}{[x' = x^3][x' := x^3]j(x, x') \geq 0} \\
\text{CQ} \frac{(x \cdot x)' \geq 0 \leftrightarrow j(x, x') \geq 0}{(x \cdot x \geq 1)' \leftrightarrow j(x, x') \geq 0}
\end{array}$$

Proofs based entirely on equivalences for solving differential equations involve DG for introducing a time variable, DC to cut the solutions in, DW to export the solution to the postcondition, inverse DC to remove the evolution domain constraints again, inverse DG to remove the original differential equations, and finally DS to solve the differential equation for time:

$$\begin{array}{c}
\text{R} \frac{\text{..} \rightarrow \forall s \geq 0 (x_0 + \frac{a}{2}s^2 + v_0s \geq 0)}{[t := 0 + 1s]x_0 + \frac{a}{2}t^2 + v_0t \geq 0} \\
\text{DS} \frac{\phi \rightarrow [t' = 1]x_0 + \frac{a}{2}t^2 + v_0t \geq 0}{\phi \rightarrow [v' = a, t' = 1]x_0 + \frac{a}{2}t^2 + v_0t \geq 0} \\
\text{DG} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1]x_0 + \frac{a}{2}t^2 + v_0t \geq 0}{\phi \rightarrow [x' = v, v' = a, t' = 1 \ \& \ v = v_0 + at]x_0 + \frac{a}{2}t^2 + v_0t \geq 0} \\
\text{DC} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1 \ \& \ v = v_0 + at \wedge x = x_0 + \frac{a}{2}t^2 + v_0t]x_0 + \frac{a}{2}t^2 + v_0t \geq 0}{\phi \rightarrow [x' = v, v' = a, t' = 1 \ \& \ v = v_0 + at \wedge x = x_0 + \frac{a}{2}t^2 + v_0t] (x = x_0 + \frac{a}{2}t^2 + v_0t \rightarrow x \geq 0)} \\
\text{G, K} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1 \ \& \ v = v_0 + at \wedge x = x_0 + \frac{a}{2}t^2 + v_0t]x \geq 0}{\phi \rightarrow [x' = v, v' = a, t' = 1 \ \& \ v = v_0 + at]x \geq 0} \\
\text{DW} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1]x \geq 0}{\phi \rightarrow \exists t [x' = v, v' = a, t' = 1]x \geq 0} \\
\text{DC} \frac{\phi \rightarrow [x' = v, v' = a]x \geq 0}{\phi \rightarrow [x' = v, v' = a]x \geq 0}
\end{array}$$

where ϕ is $a \geq 0 \wedge v = v_0 \geq 0 \wedge x = x_0 \geq 0$. The existential quantifier for t is instantiated by 0, leading to $[t := 0]$ (suppressed in the proof for readability reasons). The 4 uses of DC lead to 2 additional premises proving that $v = v_0 + at$ and then $x = x_0 + \frac{a}{2}t^2 + v_0t$ are differential invariants (using DI, DE, DW). Shortcuts using DW are possible but the above proof generalize to $\langle \rangle$ because it is an equivalence proof. The additional premise for DC with $v = v_0 + at$ proves as follows:

$$\begin{array}{c}
\text{US} \frac{\text{+}' \frac{*}{(f(\bar{x}) + g(\bar{x}))' = (f(\bar{x}))' + (g(\bar{x}))'}}{(v_0 + at)' = (v_0)' + (at)'} \\
\text{+}' \frac{(v_0 + at)' = (v_0)' + (at)'}{(v_0 + at)' = 0 + a(t')} \\
\text{CQ} \frac{v' = (v_0 + at)' \leftrightarrow v' = 0 + at'}{(v = v_0 + at)' \leftrightarrow v' = 0 + at'} \\
\text{R} \frac{*}{a = 0 + a \cdot 1} \\
\text{[':=]} \frac{[v' := a][t' := 1]v' = 0 + at'}{[v' := a][t' := 1]v' = 0 + at'} \\
\text{CE} \frac{[v' := a][t' := 1](v = v_0 + at)'}{[v' := a][t' := 1](v = v_0 + at)'} \\
\text{G} \frac{[x' = v, v' = a, t' = 1][v' := a][t' := 1](v = v_0 + at)'}{[x' = v, v' = a, t' = 1](v = v_0 + at)'} \\
\text{DE} \frac{[x' = v, v' = a, t' = 1](v = v_0 + at)'}{[x' = v, v' = a, t' = 1](v = v_0 + at)'} \\
\text{DI} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1]v = v_0 + at}{\phi \rightarrow [x' = v, v' = a, t' = 1]v = v_0 + at}
\end{array}$$

The additional premise for DC with $x = x_0 + \frac{a}{2}t^2 + v_0t$ proves as follows:

$$\begin{array}{c}
\text{R} \frac{*}{v = v_0 + at \rightarrow v = at \cdot 1 + v_0 \cdot 1} \\
\text{[':=]} \frac{[v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t']}{[v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t']} \\
\text{CE} \frac{v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t'}{v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t'} \\
\text{G} \frac{[x' = v, v' = a, t' = 1 \& v = v_0 + at](v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t')}{[x' = v, v' = a, t' = 1 \& v = v_0 + at](v = v_0 + at \rightarrow [x' := v][t' := 1]x' = att' + v_0t')} \\
\text{DW} \frac{[x' = v, v' = a, t' = 1 \& v = v_0 + at][x' := v][t' := 1]x' = att' + v_0t'}{[x' = v, v' = a, t' = 1 \& v = v_0 + at]x' = att' + v_0t'} \\
\text{DE} \frac{[x' = v, v' = a, t' = 1 \& v = v_0 + at]x' = att' + v_0t'}{[x' = v, v' = a, t' = 1 \& v = v_0 + at]x' = att' + v_0t'} \\
\text{DI} \frac{\phi \rightarrow [x' = v, v' = a, t' = 1 \& v = v_0 + at]x' = att' + v_0t'}{\phi \rightarrow [x' = v, v' = a, t' = 1 \& v = v_0 + at]x' = att' + v_0t'}
\end{array}$$

5.2 Differential Substitution Lemmas

The key insight for the soundness of DI is that the analytic time-derivative of the value of a term η along a differential equation $x' = \theta \& \psi$ agrees with the values of its differential $(\eta)'$ along the vector field of that differential equation.

Lemma 11 (Differential lemma). *If $I, \varphi \models x' = \theta \wedge \psi$ holds for some flow $\varphi : [0, r] \rightarrow \mathcal{S}$ of any duration $r > 0$, then for all $0 \leq \zeta \leq r$:*

$$[(\eta)']^I \varphi(\zeta) = \frac{d[\eta]^I \varphi(t)}{dt}(\zeta)$$

Proof. By chain rule [13, §3.10]:

$$\frac{d[\eta]^I \varphi(t)}{dt}(\zeta) = ([\eta]^I \circ \varphi)'(\zeta) = (\nabla[\eta]^I)(\varphi(\zeta)) \cdot \varphi'(\zeta) = \sum_x \frac{\partial[\eta]^I}{\partial x}(\varphi(\zeta)) \varphi'(\zeta)(x)$$

where $(\nabla[\eta]^I)(\varphi(\zeta))$, the spatial gradient $\nabla[\eta]^I$ at $\varphi(\zeta)$, is the vector of $\frac{\partial[\eta]^I}{\partial x}(\varphi(\zeta)) = \frac{\partial[\eta]^I \varphi(\zeta)_x^X}{\partial X}$. Chain rule and Def. 4 and Def. 6, thus, imply:

$$[(\eta)']^I \varphi(\zeta) = \sum_x \varphi(\zeta)(x') \frac{\partial[\eta]^I \varphi(\zeta)_x^X}{\partial X} = \sum_x \frac{\partial[\eta]^I \varphi(\zeta)_x^X}{\partial X} \frac{d\varphi(t)(x)}{dt}(\zeta) = \frac{d[\eta]^I \varphi(t)}{dt}(\zeta)$$

□

The key insight for the soundness of differential effects DE is that differential assignments mimicking the differential equation are vacuous along that differential equation. The differential substitution resulting from a subsequent use of $[\cdot :=]$ is crucial to relay the values of the time-derivatives of the state variables x along a differential equation by way of their corresponding differential symbol x' . In combination, this makes it possible to soundly substitute the right-hand side of a differential equation for its left-hand side in a proof.

Lemma 12 (Differential assignment). *If $I, \varphi \models x' = \theta \wedge \psi$ for some flow $\varphi : [0, r] \rightarrow \mathcal{S}$ of any duration $r \geq 0$, then*

$$I, \varphi \models \phi \leftrightarrow [x' := \theta]\phi$$

Proof. $I, \varphi \models x' = \theta \wedge \psi$ implies $\varphi(\zeta) \in \llbracket x' = \theta \wedge \psi \rrbracket^I$, i.e. $\varphi(\zeta)(x') = \llbracket \theta \rrbracket^I \varphi(\zeta)$ and $\varphi(\zeta) \in \llbracket \psi \rrbracket^I$ for all $0 \leq \zeta \leq r$. Consequently $(\varphi(\zeta), \varphi(\zeta)) \in \llbracket x' := \theta \rrbracket^I$ does not change the state, so that ϕ and $[x' := \theta]\phi$ are equivalent along φ . \square

The final insights for differential invariant reasoning for differential equations are syntactic ways of computing differentials, which can be internalized as axioms $(+', \cdot', \circ')$, since differentials are syntactically represented in differential-form $\mathbf{d}\mathcal{L}$.

Lemma 13 (Derivations). *The following equations of differentials are valid:*

$$(f)' = 0 \quad \text{for arity 0 functions/numbers } f \quad (2)$$

$$(x)' = x' \quad \text{for variables } x \in \mathcal{V} \quad (3)$$

$$(\theta + \eta)' = (\theta)' + (\eta)' \quad (4)$$

$$(\theta \cdot \eta)' = (\theta)' \cdot \eta + \theta \cdot (\eta)' \quad (5)$$

$$[y := \theta][y' := 1]((f(\theta))' = (f(y))' \cdot (\theta)') \quad \text{for } y, y' \notin \theta \quad (6)$$

Proof. The proof shows each equation separately. The first parts consider any constant function (i.e. arity 0) or number literal f for (2) and align the differential $(x)'$ of a term that happens to be a variable $x \in \mathcal{V}$ with its corresponding differential symbol $x' \in \mathcal{V}'$ for (3). The other cases exploit linearity for (4) and Leibniz properties of partial derivatives for (5). Case (6) exploits the chain rule and assignments and differential assignments for the fresh y, y' to mimic partial derivatives. Equation (6) generalizes to functions f of arity $n > 1$, in which case \cdot is the (definable) Euclidean scalar product.

$$\llbracket (f)' \rrbracket^I \nu = \sum_x \nu(x') \frac{\partial \llbracket f \rrbracket^I \nu_x^X}{\partial X} = \sum_x \nu(x') \frac{\partial I(f)}{\partial X} = 0 \quad (2)$$

$$\llbracket (x)' \rrbracket^I \nu = \sum_y \nu(y') \frac{\partial \llbracket x \rrbracket^I \nu_y^X}{\partial X} = \nu(x') \frac{\partial \llbracket x \rrbracket^I \nu_x^X}{\partial X} = \nu(x') \frac{\partial X}{\partial X} = \nu(x') = \llbracket x' \rrbracket^I \nu \quad (3)$$

$$\begin{aligned} \llbracket (\theta + \eta)' \rrbracket^I \nu &= \sum_x \nu(x') \frac{\partial \llbracket \theta + \eta \rrbracket^I \nu_x^X}{\partial X} = \sum_x \nu(x') \frac{\partial (\llbracket \theta \rrbracket^I \nu_x^X + \llbracket \eta \rrbracket^I \nu_x^X)}{\partial X} \\ &= \sum_x \nu(x') \left(\frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} + \frac{\partial \llbracket \eta \rrbracket^I \nu_x^X}{\partial X} \right) \end{aligned}$$

$$\begin{aligned}
&= \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} + \sum_x \nu(x') \frac{\partial \llbracket \eta \rrbracket^I \nu_x^X}{\partial X} \\
&= \llbracket (\theta)' \rrbracket^I \nu + \llbracket (\eta)' \rrbracket^I \nu = \llbracket (\theta)' + (\eta)' \rrbracket^I \nu \tag{4} \\
\llbracket (\theta \cdot \eta)' \rrbracket^I \nu &= \sum_x \nu(x') \frac{\partial \llbracket \theta \cdot \eta \rrbracket^I \nu_x^X}{\partial X} = \sum_x \nu(x') \frac{\partial (\llbracket \theta \rrbracket^I \nu_x^X \cdot \llbracket \eta \rrbracket^I \nu_x^X)}{\partial X} \\
&= \sum_x \nu(x') \left(\llbracket \eta \rrbracket^I \nu \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} + \llbracket \theta \rrbracket^I \nu \frac{\partial \llbracket \eta \rrbracket^I \nu_x^X}{\partial X} \right) \\
&= \llbracket \eta \rrbracket^I \nu \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I \nu_x^X}{\partial X} + \llbracket \theta \rrbracket^I \nu \sum_x \nu(x') \frac{\partial \llbracket \eta \rrbracket^I \nu_x^X}{\partial X} \\
&= \llbracket (\theta)' \rrbracket^I \nu \cdot \llbracket \eta \rrbracket^I \nu + \llbracket \theta \rrbracket^I \nu \cdot \llbracket (\eta)' \rrbracket^I \nu = \llbracket (\theta)' \cdot \eta + \theta \cdot (\eta)' \rrbracket^I \nu \tag{5}
\end{aligned}$$

Proving that $\nu \in \llbracket [y := \theta][y' := 1]((f(\theta))' = (f(y))' \cdot (\theta)') \rrbracket^I$ requires showing that $\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1} \in \llbracket (f(\theta))' = (f(y))' \cdot (\theta)' \rrbracket^I$, i.e. $\llbracket (f(\theta))' \rrbracket^I \nu_y^{\llbracket \theta \rrbracket^I \nu_1^1} = \llbracket (f(y))' \cdot (\theta)' \rrbracket^I \nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}$. This is equivalent to $\llbracket (f(\theta))' \rrbracket^I \nu = \llbracket (f(y))' \rrbracket^I \nu_y^{\llbracket \theta \rrbracket^I \nu_1^1} \cdot \llbracket (\theta)' \rrbracket^I \nu$ by Lemma 2 since $\nu = \nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}$ on $\{y, y'\}^c$ and $y, y' \notin \text{FV}(\theta)$ by assumption, so $y, y' \notin \text{FV}((f(\theta))')$ and $y, y' \notin \text{FV}((\theta)')$. The latter equation proves using the chain rule and a fresh variable z when denoting $\llbracket f \rrbracket^I \stackrel{\text{def}}{=} I(f)$:

$$\begin{aligned}
\llbracket (f(\theta))' \rrbracket^I \nu &= \sum_x \nu(x') \frac{\partial \llbracket f(\theta) \rrbracket^I}{\partial x}(\nu) = \sum_x \nu(x') \frac{\partial (\llbracket f \rrbracket^I \circ \llbracket \theta \rrbracket^I)}{\partial x}(\nu) \\
&\stackrel{\text{chain}}{=} \sum_x \nu(x') \frac{\partial \llbracket f \rrbracket^I}{\partial y}(\llbracket \theta \rrbracket^I \nu) \cdot \frac{\partial \llbracket \theta \rrbracket^I}{\partial x}(\nu) \\
&= \frac{\partial \llbracket f \rrbracket^I}{\partial y}(\llbracket \theta \rrbracket^I \nu) \cdot \sum_x \nu(x') \frac{\partial \llbracket \theta \rrbracket^I}{\partial x}(\nu) = \frac{\partial \llbracket f \rrbracket^I}{\partial y}(\llbracket \theta \rrbracket^I \nu) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \frac{\partial I(f)}{\partial y}(\llbracket \theta \rrbracket^I \nu) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \frac{\partial I(f)}{\partial z}(\llbracket \theta \rrbracket^I \nu) 1 \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \frac{\partial I(f)}{\partial z}(\llbracket y \rrbracket^I \nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}) \frac{\partial \llbracket y \rrbracket^I}{\partial y}(\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&\stackrel{\text{chain}}{=} \frac{\partial (I(f) \circ \llbracket y \rrbracket^I)}{\partial y}(\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \left(\frac{\partial \llbracket f(y) \rrbracket^I}{\partial y}(\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}) \right) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \left(\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}(y') \frac{\partial \llbracket f(y) \rrbracket^I}{\partial y}(\nu_y^{\llbracket \theta \rrbracket^I \nu_1^1}) \right) \cdot \llbracket (\theta)' \rrbracket^I \nu
\end{aligned}$$

$$\begin{aligned}
&= \left(\sum_{x \in \{y\}} \nu_y^{\llbracket \theta \rrbracket^I \nu_{y'}^1}(x') \frac{\partial \llbracket f(y) \rrbracket^I}{\partial x} (\nu_y^{\llbracket \theta \rrbracket^I \nu_{y'}^1}) \right) \cdot \llbracket (\theta)' \rrbracket^I \nu \\
&= \llbracket (f(y))' \rrbracket^I \nu_y^{\llbracket \theta \rrbracket^I \nu_{y'}^1} \cdot \llbracket (\theta)' \rrbracket^I \nu
\end{aligned} \tag{6}$$

□

5.3 Soundness

Theorem 14 (Soundness). *The \mathbf{dL} axioms and proof rules in Fig. 2, 3 are sound, i.e. the axioms are valid formulas and the conclusion of a rule is valid if its premises are. All US instances of the proof rules (with $FV(\sigma) = \emptyset$) are sound.*

Proof. The axioms (and most proof rules) in Fig. 2 are special instances of corresponding axiom schemata and proof rules for differential dynamic logic [7] and, thus, sound. All proof rules except US are even *locally sound*, i.e. for all I : if all their premises ϕ_j are valid in I ($I \models \phi_j$) then their conclusion ψ is, too ($I \models \psi$). Local soundness implies soundness. In addition, local soundness implies that US can be used to soundly instantiate proof rules just like it soundly instantiates axioms (Theorem 10). If

$$\frac{\phi_1 \quad \dots \quad \phi_n}{\psi} \tag{7}$$

is a locally sound proof rule then its substitution instance is locally sound:

$$\frac{\sigma(\phi_1) \quad \dots \quad \sigma(\phi_n)}{\sigma(\psi)} \tag{8}$$

where σ is any uniform substitution (for which the above results are defined, i.e. no clash) with $FV(\sigma) = \emptyset$. To show this, consider any I in which all premises of (8) are valid, i.e. $I \models \sigma(\phi_j)$ for all j . That is, $\nu \in \llbracket \sigma(\phi_j) \rrbracket^I$ for all ν and all j . By Lemma 8, $\nu \in \llbracket \sigma(\phi_j) \rrbracket^I$ is equivalent to $\nu \in \llbracket \phi_j \rrbracket^{\sigma_\nu^* I}$, which, thus, also holds for all ν and all j . By Corollary 6, $\llbracket \phi_j \rrbracket^{\sigma_\nu^* I} = \llbracket \phi_j \rrbracket^{\sigma_\omega^* I}$ for any ω , since $FV(\sigma) = \emptyset$. Consequently, all premises of (7) are valid in $\sigma_\omega^* I$, i.e. $\sigma_\omega^* I \models \phi_j$ for all j . Thus, $\sigma_\omega^* I \models \psi$ by local soundness of (7). That is, $\nu \in \llbracket \psi \rrbracket^{\sigma_\nu^* I} = \llbracket \psi \rrbracket^{\sigma_\omega^* I}$ by Corollary 6 for all ν . By Lemma 8, $\nu \in \llbracket \psi \rrbracket^{\sigma_\nu^* I}$ is equivalent to $\nu \in \llbracket \sigma(\psi) \rrbracket^I$, which continues to hold for all ν . Thus, $I \models \sigma(\psi)$, i.e. the conclusion of (8) is valid in I , hence (8) locally sound. Consequently, all US instances of the locally sound proof rules of \mathbf{dL} with $FV(\sigma) = \emptyset$ are locally sound. Note that \forall, MP can be augmented soundly to use $p(\bar{x})$ instead of $p(x)$ or p , respectively, such that the $FV(\sigma) = \emptyset$ requirement will be met during US instances of all rules.

DW Soundness of DW uses that differential equations can never leave their evolution domain by Def. 6. To show $\nu \in \llbracket [x' = f(x) \ \& \ q(x)]q(x) \rrbracket^I$, consider any φ of any duration $r \geq 0$ solving $I, \varphi \models x' = f(x) \wedge q(x)$. Then $I, \varphi \models q(x)$ hence $\varphi(r) \in \llbracket q(x) \rrbracket^I$.

- DC** Soundness of DC is a stronger version of soundness for the differential cut rule [6]. DC is a differential version of the modal modus ponens K. The core is that if $r(x)$ always holds after the differential equation and $p(x)$ always holds after the differential equation $x' = f(x) \& q(x) \wedge r(x)$ that is restricted to $r(x)$, then $p(x)$ always holds after the differential equation $x' = f(x) \& q(x)$ without that additional restriction. Let $\nu \in \llbracket [x' = f(x) \& q(x)]r(x) \rrbracket^I$. Since all restrictions of solutions are solutions, this is equivalent to $I, \varphi \models r(x)$ for all φ of any duration solving $I, \varphi \models x' = f(x) \wedge q(x)$ and starting in $\varphi(0) = \nu$ on $\{x'\}^{\mathbb{C}}$. Consequently, for all φ starting in $\varphi(0) = \nu$ on $\{x'\}^{\mathbb{C}}$: $I, \varphi \models x' = f(x) \wedge q(x)$ is equivalent to $I, \varphi \models x' = f(x) \wedge q(x) \wedge r(x)$. Hence, $\nu \in \llbracket [x' = f(x) \& q(x) \wedge r(x)]p(x) \rrbracket^I$ is equivalent to $\nu \in \llbracket [x' = f(x) \& q(x)]p(x) \rrbracket^I$.
- DE** Soundness of DE is genuine to differential-form \mathbf{dL} leveraging Lemma 12. Consider any state ν . Then $\nu \in \llbracket [x' = f(x) \& q(x)]p(x, x') \rrbracket^I$ iff $\varphi(r) \in \llbracket p(x, x') \rrbracket^I$ for all solutions $\varphi : [0, r] \rightarrow \mathcal{S}$ of $I, \varphi \models x' = f(x) \wedge q(x)$ of any duration r starting in $\varphi(0) = \nu$ on $\{x'\}^{\mathbb{C}}$. That is equivalent to: for all φ , if $I, \varphi \models x' = f(x) \wedge q(x)$ then $I, \varphi \models p(x, x')$. By Lemma 12, $I, \varphi \models p(x, x')$ iff $I, \varphi \models [x' := f(x)]p(x, x')$, so, that is equivalent to $\varphi(r) \in \llbracket [x' := f(x)]p(x, x') \rrbracket^I$ for all solutions $\varphi : [0, r] \rightarrow \mathcal{S}$ of $I, \varphi \models x' = f(x) \wedge q(x)$ of any duration r starting in $\varphi(0) = \nu$ on $\{x'\}^{\mathbb{C}}$, which is, consequently, equivalent to $\nu \in \llbracket [x' = f(x) \& q(x)][x' := f(x)]p(x, x') \rrbracket^I$.
- DI** Soundness of DI has some relation to the soundness proof for differential invariants [6], yet is generalized to leverage differentials. The proof is only shown for $p(x) \stackrel{\text{def}}{=} g(x) \geq 0$, in which case $(p(x))' \equiv ((g(x)))' \geq 0$. Consider a state ν in which $\nu \in \llbracket q(x) \rightarrow (p(x) \wedge [x' = f(x) \& q(x)](p(x)))' \rrbracket^I$. If $\nu \notin \llbracket q(x) \rrbracket^I$, there is nothing to show, because there is no solution of $x' = f(x) \& q(x)$ for any duration, so the consequence holds vacuously. Otherwise, $\nu \in \llbracket q(x) \rrbracket^I$ implies $\nu \in \llbracket p(x) \wedge [x' = f(x) \& q(x)](p(x))' \rrbracket^I$. To show that $\nu \in \llbracket [x' = f(x) \& q(x)]p(x) \rrbracket^I$ consider any solution φ of any duration $r \geq 0$. The case $r = 0$ follows from $\nu \in \llbracket p(x) \rrbracket^I$ by Lemma 3 since $\text{FV}(p(x)) = \{x\}$ is disjoint from $\{x'\}$, which is changed by evolutions of any duration. That leaves the case $r > 0$. Let $I, \varphi \models x' = f(x) \& q(x)$, which, by $\nu \in \llbracket [x' = f(x) \& q(x)](p(x))' \rrbracket^I$, implies $I, \varphi \models (p(x))'$. Since $r > 0$, Lemma 11 implies $0 \leq ((g(x)))' \varphi(\zeta) = \frac{d\llbracket g(x) \rrbracket^I \varphi(t)}{dt}(\zeta)$ for all ζ . Together with $\varphi(0) \in \llbracket p(x) \rrbracket^I$ (by Lemma 3 and $\text{FV}(p(x)) \cap \{x'\} = \emptyset$), i.e. $\varphi(0) \in \llbracket g(x) \geq 0 \rrbracket^I$, this implies $\varphi(\zeta) \in \llbracket g(x) \geq 0 \rrbracket^I$ for all ζ , including r , by the mean-value theorem since $\llbracket g(x) \rrbracket^I \varphi(t)$ is continuous in t on $[0, r]$ and differentiable on $(0, r)$.
- DG** Soundness of DG is a constructive variation of the soundness proof for differential auxiliaries [8]. Let $\nu \in \llbracket \exists y [x' = f(x), y' = a(x)y + b(x) \& q(x)]p(x) \rrbracket^I$, that is, $\nu_x^d \in \llbracket [x' = f(x), y' = a(x)y + b(x) \& q(x)]p(x) \rrbracket^I$ for some d . In order to show that $\nu \in \llbracket [x' = f(x) \& q(x)]p(x) \rrbracket^I$, consider any $\varphi : [0, r] \rightarrow \mathcal{S}$ such that $I, \varphi \models x' = f(x) \wedge q(x)$ and $\varphi(0) = \nu$ on $\{x'\}^{\mathbb{C}}$. By modifying the values of y, y' along φ , this function can be augmented to a solution $\tilde{\varphi} : [0, r] \rightarrow \mathcal{S}$ such that $I, \tilde{\varphi} \models x' = f(x) \wedge y' = a(x)y + b(x) \wedge q(x)$ and $\tilde{\varphi}(0)(y) = d$. The assumption then implies $\tilde{\varphi}(r) \in \llbracket p(x) \rrbracket^I$, which, by Lemma 3, is equivalent to $\varphi(r) \in \llbracket p(x) \rrbracket^I$ since $y, y' \notin \text{FV}(p(x))$ and $\varphi(r) = \tilde{\varphi}(r)$ on $\{y, y'\}^{\mathbb{C}}$, which

implies $\nu \in \llbracket [x' = f(x) \& q(x)]p(x) \rrbracket^I$, since φ was arbitrary. The construction of the modification $\tilde{\varphi}$ of φ on $\{y, y'\}$ proceeds as follows. By Picard-Lindelöf [14, §10.VII], there is a solution $y : [0, r] \rightarrow \mathbb{R}$ of the initial-value problem

$$\begin{aligned} y(0) &= d \\ y'(t) &= F(t, y(t)) \stackrel{\text{def}}{=} y(t) \llbracket a(x) \rrbracket^I \varphi(t) + \llbracket b(x) \rrbracket^I \varphi(t) \end{aligned} \quad (9)$$

because $F(t, y)$ is continuous on $[0, r] \times \mathbb{R}$ (since $\llbracket a(x) \rrbracket^I \varphi(t)$ and $\llbracket b(x) \rrbracket^I \varphi(t)$ are continuous in t as compositions of the, by Def. 4 smooth, evaluation function and the continuous solution $\varphi(t)$ of a differential equation) and because $F(t, y)$ satisfies the Lipschitz condition

$$\|F(t, y) - F(t, z)\| = \|(y - z) \llbracket a(x) \rrbracket^I \varphi(t)\| \leq \|y - z\| \max_{t \in [0, r]} \llbracket a(x) \rrbracket^I \varphi(t)$$

where the maximum exists, because it is a maximum of a continuous function on the compact set $[0, r]$. The modification $\tilde{\varphi}$ agrees with φ on $\{y, y'\}^c$ and is defined as $\tilde{\varphi}(t)(y) = y(t)$ and $\tilde{\varphi}(t)(y') = F(t, y(t)) = y'(t)$ on $\{y, y'\}$, respectively, for the solution $y(t)$ of (9). By construction, $\tilde{\varphi}(0)(y) = d$ and $I, \tilde{\varphi} \models x' = f(x) \wedge y' = a(x)y + b(x) \wedge q(x)$, because $\varphi(t) = \tilde{\varphi}(t)$ on $\{y, y'\}^c$ so that $x' = f(x) \& q(x)$ continues to hold along $\tilde{\varphi}$ by Lemma 2 because $y, y' \notin \text{FV}(x' = f(x) \& q(x))$, and because $y' = a(x)y + b(x)$ holds along $\tilde{\varphi}$ by (9).

Conversely, let $\nu \in \llbracket [x' = f(x) \& q(x)]p(x) \rrbracket^I$. This direction shows a stronger version of $\nu \in \llbracket \exists y [x' = f(x), y' = a(x)y + b(x) \& q(x)]p(x) \rrbracket^I$ by showing that

$\nu_y^d \in \llbracket [x' = f(x), y' = \eta \& q(x)]p(x) \rrbracket^I$ for all $d \in \mathbb{R}$ and all terms η . Consider any $\varphi : [0, r] \rightarrow \mathcal{S}$ such that $I, \varphi \models x' = f(x) \wedge y' = \eta \wedge q(x)$ with $\varphi(0) = \nu_y^d$ on $\{x', y'\}^c$. Then the restriction $\varphi|_{\{y, y'\}^c}$ of φ to $\{y, y'\}^c$ with $\varphi|_{\{y, y'\}^c}(t) = \nu_y^d$ on $\{y, y'\}$ for all $t \in [0, r]$ still solves $I, \varphi|_{\{y, y'\}^c} \models x' = f(x) \wedge q(x)$ by Lemma 2 since $\varphi|_{\{y, y'\}^c} = \varphi$ on $\{y, y'\}^c$ and $y, y' \notin \text{FV}(x' = f(x) \& q(x))$. It also satisfies $\varphi|_{\{y, y'\}^c}(0) = \nu_y^d$ on $\{x'\}^c$, because $\varphi(0) = \nu_y^d$ on $\{x', y'\}^c$ yet $\varphi|_{\{y, y'\}^c}(t)(y') = \nu_y^d(y')$. Thus, by assumption, $\varphi|_{\{y, y'\}^c}(r) \in \llbracket p(x) \rrbracket^I$, which implies $\varphi(r) \in \llbracket p(x) \rrbracket^I$ by Lemma 3, because $\varphi = \varphi|_{\{y, y'\}^c}$ on $\{y, y'\}^c$ and $y, y' \notin \text{FV}(p(x))$,

DS Soundness of the solution axiom DS follows from existence and uniqueness of global solutions of constant differential equations. Consider any state ν . There is a unique [14, §10.VII] global solution $\varphi : [0, \infty) \rightarrow \mathcal{S}$ defined as $\varphi(\zeta)(x) \stackrel{\text{def}}{=} \llbracket x + f t \rrbracket^I \nu_t^\zeta$ and $\varphi(\zeta)(x') \stackrel{\text{def}}{=} \frac{d\varphi(t)(x)}{dt}(\zeta) = I(f)$ and $\varphi(\zeta) = \nu$ on $\{x, x'\}^c$. This solution satisfies $\varphi(0) = \nu(x)$ on $\{x'\}^c$ and $I, \varphi \models x' = f$, i.e. $\varphi(\zeta) \in \llbracket x' = f \rrbracket^I$ for all $0 \leq \zeta \leq r$. All solutions of $x' = f$ from initial state ν are restrictions of φ to subintervals of $[0, \infty)$. The (unique) state ω that satisfies $(\nu_t^\zeta, \omega) \in \llbracket x := x + f t \rrbracket^I$ agrees with $\omega = \varphi(\zeta)$ on $\{x'\}^c$, so that, by $x' \notin \text{FV}(p(x))$, Lemma 3 implies that $\omega \in \llbracket p(x) \rrbracket^I$ iff $\varphi(\zeta) \in \llbracket p(x) \rrbracket^I$.

First consider axiom $[x' = f]p(x) \leftrightarrow \forall t \geq 0 [x := x + f t]p(x)$ for $q(x) \equiv \text{true}$. If $\nu \in \llbracket [x' = f]p(x) \rrbracket^I$, then $\varphi(\zeta) \in \llbracket p(x) \rrbracket^I$ for all $\zeta \geq 0$, because the restriction of φ to $[0, \zeta)$ solves $x' = f$ from ν , thus $\omega \in \llbracket p(x) \rrbracket^I$, which implies $\nu_t^\zeta \in \llbracket [x := x + f t]p(x) \rrbracket^I$, so $\nu \in \llbracket \forall t \geq 0 [x := x + f t]p(x) \rrbracket^I$ as $\zeta \geq 0$ was arbitrary.

Conversely, $\nu \in \llbracket \forall t \geq 0 [x := x + ft]p(x) \rrbracket^I$ implies $\nu_t^\zeta \in \llbracket [x := x + ft]p(x) \rrbracket^I$ for all $\zeta \geq 0$, i.e. $\omega \in \llbracket p(x) \rrbracket^I$ when $(\nu_t^\zeta, \omega) \in \llbracket x := x + ft \rrbracket^I$. Lemma 3 again implies $\varphi(\zeta) \in \llbracket p(x) \rrbracket^I$ for all $\zeta \geq 0$, so $\nu \in \llbracket [x' = f]p(x) \rrbracket^I$, since all solutions are restrictions of φ .

Soundness of DS now follows using that all solutions $\varphi : [0, r] \rightarrow \mathcal{S}$ of $x' = f(x) \ \& \ q(x)$ satisfy $\varphi(\zeta) \in \llbracket q(x) \rrbracket^I$ for all $0 \leq \zeta \leq r$, which, using Lemma 3 as above, is equivalent to $\nu \in \llbracket \forall 0 \leq s \leq t q(x + fs) \rrbracket^I$ when $\nu(t) = r$.

[':=] Soundness of [':=] follows from the semantics of differential assignments (Def. 6) and compositionality. In detail: $x' := f$ changes the value of symbol x' to the value of f . The predicate p has the same value for arguments x' and f that have the same value.

+',',\circ' Soundness of the derivation axioms +',',\circ' follows from Lemma 13, since they are special instances of (4) and (5) and (6), respectively. For \circ' observe that $y, y' \notin g(x)$.

G Let the premise $p(\bar{x})$ be valid in some I , i.e. $I \models p(\bar{x})$, i.e. $\omega \in \llbracket p(\bar{x}) \rrbracket^I$ for all ω . Then, the conclusion $[a]p(\bar{x})$ is valid in the same I , i.e. $\nu \in \llbracket [a]p(\bar{x}) \rrbracket^I$ for all ν , because $\omega \in \llbracket p(\bar{x}) \rrbracket^I$ for all ω , so also for all ω with $(\nu, \omega) \in \llbracket a \rrbracket^I$. Thus, G is locally sound.

\forall Let the premise $p(x)$ be valid in some I , i.e. $I \models p(x)$, i.e. $\omega \in \llbracket p(x) \rrbracket^I$ for all ω . Then, the conclusion $\forall x p(x)$ is valid in the same I , i.e. $\nu \in \llbracket \forall x p(x) \rrbracket^I$ for all ν , i.e. $\nu_x^d \in \llbracket p(x) \rrbracket^I$ for all $d \in \mathbb{R}$, because $\omega \in \llbracket p(x) \rrbracket^I$ for all ω , so in particular for all $\omega = \nu_x^d$ for any $d \in \mathbb{R}$. Thus, \forall is locally sound.

CQ Let the premise $f(\bar{x}) = g(\bar{x})$ be valid in some I , i.e. $I \models f(\bar{x}) = g(\bar{x})$, i.e. $\nu \in \llbracket f(\bar{x}) = g(\bar{x}) \rrbracket^I$ for all ν , i.e. $\llbracket f(\bar{x}) \rrbracket^I \nu = \llbracket g(\bar{x}) \rrbracket^I \nu$ for all ν . Consequently, $\llbracket f(\bar{x}) \rrbracket^I \nu \in I(p)$ iff $\llbracket g(\bar{x}) \rrbracket^I \nu \in I(p)$. So, $I \models p(f(\bar{x})) \leftrightarrow p(g(\bar{x}))$. Thus, CQ is locally sound.

CE Let the premise $p(\bar{x}) \leftrightarrow q(\bar{x})$ be valid in some I , i.e. $I \models p(\bar{x}) \leftrightarrow q(\bar{x})$, i.e. $\nu \in \llbracket p(\bar{x}) \leftrightarrow q(\bar{x}) \rrbracket^I$ for all ν . Consequently, $\llbracket p(\bar{x}) \rrbracket^I = \llbracket q(\bar{x}) \rrbracket^I$. Thus, $\llbracket C(p(\bar{x})) \rrbracket^I = I(C)(\llbracket p(\bar{x}) \rrbracket^I) = I(C)(\llbracket q(\bar{x}) \rrbracket^I) = \llbracket C(q(\bar{x})) \rrbracket^I$. This implies $I \models C(p(\bar{x})) \leftrightarrow C(q(\bar{x}))$, hence the conclusion is valid in I . Thus, CE is locally sound.

CT Rule CT is a (locally sound) derived rule and only included for comparison. CT is derivable from CQ using $p(\cdot) \stackrel{\text{def}}{=} (c(\cdot) = c(g(\bar{x})))$ and reflexivity of $=$.

MP Modus ponens MP is locally sound with respect to the interpretation I and the state ν , which implies local soundness and thus soundness. If $\nu \in \llbracket p \rightarrow q \rrbracket^I$ and $\nu \in \llbracket p \rrbracket^I$ then $\nu \in \llbracket q \rrbracket^I$.

US Uniform substitution is sound by Theorem 10, just not necessarily locally sound.

□

6 Conclusions

With differential forms for local reasoning about differential equations, uniform substitutions lead to a simple and modular proof calculus for differential dynamic logic that is entirely based on axioms and axiomatic rules, instead of soundness-critical schema variables with side-conditions in axiom schemata. The US calculus is straightforward to implement and enables flexible reasoning with axioms by contextual equivalence. Efficiency can be regained by tactics that combine multiple axioms and rebalance the proof to obtain short proof search branches. Contextual equivalence rewriting for implications is possible when adding monotone quantifiers C whose substitution instances limit $_$ to positive polarity.

Acknowledgment. I thank the anonymous reviewers of the conference version [10] for their helpful feedback.

This material is based upon work supported by the National Science Foundation by NSF CAREER Award CNS-1054246. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of any sponsoring institution or government.

A Appendix

This appendix briefly discusses generalized uses and forms of the differential ghost axioms and how it generalizes the differential auxiliaries proof rule [8].

Differential Lipschitz Ghosts The differential ghost axiom DG generalizes to arbitrary Lipschitz-continuous differential equations $y' = g(x, y)$:

$$(DG_\ell) \quad \begin{aligned} &([x' = f(x) \ \& \ q(x)]p(x) \leftrightarrow \exists y [x' = f(x), y' = g(x, y) \ \& \ q(x)]p(x)) \\ &\leftarrow \exists \ell \forall x, y, z |g(x, y) - g(x, z)| \leq \ell |y - z| \end{aligned}$$

The soundness argument for DG_ℓ is an extension of the soundness proof for DG. The direction “ \leftarrow ” of DG is sound for all differential equations. The proof for the direction “ \rightarrow ” extends the proof for DG with an adaptation of the function F from (9) to the differential equation $y' = g(x, y)$:

$$\begin{aligned} y(0) &= d \\ y'(t) &= F(t, y(t)) \stackrel{\text{def}}{=} \llbracket g(x, y) \rrbracket^I \varphi(t)_y^{y(t)} \end{aligned} \tag{10}$$

This function $F(t, \delta)$ is still continuous on $[0, r] \times \mathbb{R}$ since it is a composition of the continuous evaluation (of the, by assumption, continuous term $g(x, y)$) with the (continuous) composition of the continuous function $\varphi(t)$ of t with the continuous modification of the value of variable y to δ . By assumption $F(t, y)$ is Lipschitz in y , since there is an $\ell \in \mathbb{R}$ such that for all $t, a, b \in \mathbb{R}$:

$$\begin{aligned}
|F(t, a) - F(t, b)| &= |\llbracket g(x, y) \rrbracket^I \varphi(t)_y^a - \llbracket g(x, y) \rrbracket^I \varphi(t)_y^b| = |\llbracket g(x, y) - g(x, z) \rrbracket^I \varphi(t)_{yz}^{ab}| \\
&= \underbrace{|\llbracket g(x, y) - g(x, z) \rrbracket^I \varphi(t)_{yz}^{ab}|}_{\leq \ell \llbracket y - z \rrbracket^I \varphi(t)_{yz}^{ab}} \leq \ell |a - b|
\end{aligned}$$

This establishes the only two properties of F that the soundness proof of DG was based on. The existence of a solution $y : [0, r] \rightarrow \mathbb{R}$ of (10) is, thus, established again by Picard-Lindelöf as needed for the soundness proof.

Differential Auxiliaries Rule The differential auxiliaries proof rule [8] is derivable from DG and monotonicity M.

$$(DA) \frac{p(x) \leftrightarrow \exists y r(x, y) \quad r(x, y) \rightarrow [x' = f(x), y' = g(x, y) \& q(x)] r(x, y)}{p(x) \rightarrow [x' = f(x) \& q(x)] p(x)}$$

where y is new and $y' = g(x, y)$, $y(0) = y_0$ has a solution $y : [0, \infty) \rightarrow \mathbb{R}^n$ for each y_0 .

The derivation proceeds as follows (the middle premise uses V_{\exists} with $y \notin p(x)$):

$$\begin{array}{c}
\frac{\exists y r(x, y) \rightarrow p(x)}{V_{\exists} \frac{r(x, y) \rightarrow p(x) \quad r(x, y) \rightarrow [x' = f(x), y' = g(x, y) \& q(x)] r(x, y)}{M \frac{r(x, y) \rightarrow [x' = f(x), y' = g(x, y) \& q(x)] p(x)}{\exists i \frac{r(x, y) \rightarrow \exists y [x' = f(x), y' = g(x, y) \& q(x)] p(x)}{DG \frac{r(x, y) \rightarrow [x' = f(x) \& q(x)] p(x)}{\exists i \frac{\exists y r(x, y) \rightarrow [x' = f(x) \& q(x)] p(x)}{cut \frac{p(x) \leftrightarrow \exists y r(x, y)}{p(x) \rightarrow [x' = f(x) \& q(x)] p(x)}}}
\end{array}$$

Using the following duals of $\forall i$ and V_{\forall} as well as monotonicity rule M [4] that derives from G, K:

$$(\exists i) \quad p(f) \rightarrow (\exists x p(x))$$

$$(V_{\exists}) \quad \exists x p \rightarrow p$$

$$(M) \quad \frac{\phi \rightarrow \psi}{[\alpha] \phi \rightarrow [\alpha] \psi}$$

References

- [1] Alonzo Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5(2):56–68, 1940.
- [2] Alonzo Church. *Introduction to Mathematical Logic, Volume I*. Princeton University Press, Princeton, NJ, 1956.
- [3] Leon Henkin. Banishing the rule of substitution for functional variables. *J. Symb. Log.*, 18(3):pp. 201–208, 1953.
- [4] André Platzer. Differential game logic. *ACM Trans. Comput. Log.* To appear. Preprint at arXiv 1408.1980.

- [5] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. doi:10.1007/s10817-008-9103-8.
- [6] André Platzer. Differential-algebraic dynamic logic for differential-algebraic programs. *J. Log. Comput.*, 20(1):309–352, 2010. doi:10.1093/logcom/exn070.
- [7] André Platzer. The complete proof theory of hybrid systems. In *LICS*, pages 541–550. IEEE, 2012. doi:10.1109/LICS.2012.64.
- [8] André Platzer. The structure of differential invariants and differential cut elimination. *Log. Meth. Comput. Sci.*, 8(4):1–38, 2012. doi:10.2168/LMCS-8(4:16)2012.
- [9] André Platzer. Differential game logic. *CoRR*, abs/1408.1980, 2014. arXiv:1408.1980.
- [10] André Platzer. A uniform substitution calculus for differential dynamic logic. In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481. Springer, 2015. doi:10.1007/978-3-319-21401-6_32.
- [11] André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008. doi:10.1007/978-3-540-71070-7_15.
- [12] H. Gordon Rice. Classes of recursively enumerable sets and their decision problems. *Trans. AMS*, 89:25–59, 1953.
- [13] Wolfgang Walter. *Analysis 2*. Springer, 4 edition, 1995.
- [14] Wolfgang Walter. *Ordinary Differential Equations*. Springer, 1998.